UNCLASSIFIED - UNLIMITED

NORTH ATLANTIC TREATY ORGANIZATION
DEFENCE RESEARCH GROUP

0103-95

TECHNICAL REPORT
AC/243(Panel 8)TR/17

COADE : A FRAMEWORK FOR COGNITIVE ANALYSIS, DESIGN AND EVALUATION

FINAL REPORT

Panel 8 on the Defence Applications of Human and Bio-Medical Sciences

RSG.19 on Decision Aids in Command and Control

UNCLASSIFIED

REPORT DOCUMENTATION PAGE

1. Recipient's Reference:	2. Further Reference:
3. Originator's Reference: AC/243(Panel 8)TR/17	4. Security Classification: UU
	5. Date: 23 January 95 — 6. Total Pages: 221

7. Title (NU):

COADE - A FRAMEWORK FOR COGNITIVE ANALYSIS, DESIGN, AND EVALUATION

8. Presented at:

9. Authors/Editors:

P. Essens(1), J. Fallesen(2), C. McCann(3) J. Cannon-Bowers(4), G. Dorfel(5)

10. Author(s)/Editor(s) Address:

1) TNO HFR Institute, NL
2) ARI, Fort Leavenworth, US
3) DCIEM, CA, 4) NAWC, US
5) FGAN/FAT, Germany

11. NATO Staff Point of Contact:

Defence Research Section
NATO Headquarters
B-1110 Brussels
Belgium
(Not a Distribution Centre)

12. Distribution Statement:

13. Keywords/Descriptors:

DECISION AIDS, DECISION SUPPORT SYSTEMS, COGNITIVE TASK ANALYSIS, P.8/RSG.19, EVALUATION, HUMAN-COMPUTER INTERACTION, HUMAN FACTORS ENGINEERING, SYSTEM DESIGN, COMMAND AND CONTROL, COGNITIVE PERFORMANCE ANALYSIS,

14. Abstract:

The development of support of decision making processes in complex systems requires a systematic approach based upon knowledge of the human's role, capabilities, and the tasks to be performed. Results from a survey and a workshop show that there is a need for methodologies that systematically address the cognitive factors of complex task situations. COADE provides the developer and cognitive specialist with an approach to the development of cognitively-centered systems. The COADE framework comprises a set of activities for cognitive analysis, design and evaluation. Analysis activities result in the specification of cognitive requirements; design activities translate those into design requirements; evaluation activities control the quality of the intermediate and final products of the development process.

CONSEIL DE L'ATLANTIQUE NORD
NORTH ATLANTIC COUNCIL

UNCLASSIFIED/UNLIMITED

ORIGINAL: ENGLISH
23 January 1995

TECHNICAL REPORT
AC/243(Panel 8)TR/17

DEFENCE RESEARCH GROUP

PANEL 8 ON THE DEFENCE APPLICATION OF HUMAN AND BIO-MEDICAL SCIENCES

Final Technical Report on COADE: A Framework for Cognitive Analysis, Design, and Evaluation

1. This is the final technical report prepared by P.8/RSG.19 on Decision Aids in Command and Control.

2. The Executive Summary ("Yellow Pages") will be distributed under reference AC/243-N/424 dated 23 January 1995.

(Signed) Dr. J. VERMOREL
Secretary, AC/243(Panel 8)

NATO,
1110 Brussels.

This page has been left blank intentionally

OUTLINE OF CHAPTERS

Executive Summary ix
Chapter 1. Introduction 1
Chapter 2. Survey of C2 Decision Aids Development 5
Chapter 3. Summary of Workshop on Draft Framework 11
Chapter 4. Introduction to the COADE Framework 15
Chapter 5. COADE Framework 23
5.1. ANALYSE 26
5.2. DESIGN 50
5.3. EVALUATE 79
Chapter 6. COADE Background 95
6.1. Models of Decision Making 97
6.2. A General Cognitive Model of C2 Decision Making 103
6.3. Cognitive Task Analysis 113
6.4. Cognitive Performance Analysis 131
6.5. Performance Aiding Strategies and Guidelines 143
6.6. Human-Computer Interface Design 153
6.7. Evaluation 161
References 165

Appendix A. Concepts used in Modelling Cognition 183

FIGURES AND TABLES

Figure 0.1. Main Components of the COADE Framework: ANALYSE, DESIGN, and EVALUATE Activities with Reference Information xi
Figure 4.1. Relationship between COADE Activities and DOD-STD-2167A System Development Phases 16
Figure 4.2. COADE Framework with Reference Information, ANALYSE, DESIGN, EVALUATE Activities and Their Products 18
Figure 5.1. COADE Model of Cognitively-centred System Design: Relationships between Activities, Products and Reference Information, ... 25
Figure 6.1. COADE Model of Cognitive Processes in C2 Decision Making 104
Figure 6.2. Levels of Abstraction in Concept and Procedural Nets 106
Figure 6.3. Schema Supported by Concept and Procedural Nets 107
Figure 6.4. COADE Model of Knowledge Representation Used in Decision Making 108
Figure 6.5. A Cognitive Goal-based Taxonomy of Performance Aiding Strategies. 145

Table 2.1. Responses from C2 Decision Aid Development Survey 5
Table 4.1. Relationship between COADE ANALYSE activities. 19
Table 5.1. Sample Comparison of C2 Measures (adapted from Garlinger & Fallesen, 1988) ...35
Table 5.2. Example measures from ACCES (adapted from Halpin, 1992) 36
Table 5.3. Decision Making Difficulties (Miller, 1971) 39
Table 5.4. Example Error Behaviours (adapted from Altman, 1967) 40
Table 5.5. Human Error Classification Scheme (Rouse & Rouse, 1983) 40
Table 5.6. Generic Error Modelling System (GEMS) (Reason, 1991) 41
Table 5.7. Cognitive Limitations 48
Table 5.8. Examples of Trade-offs between Aiding and Alternative Solutions 52
Table 5.9. Taxonomy of Performance Aiding Strategies 57
Table 5.10. Aiding Guidelines 60
Table 6.1. Part 1. Knowledge Elicitation Techniques: Technique, Description, Representation, Sources 118
Table 6.1. Part 2. Knowledge Elicitation Techniques: Technique, Strength, Limitations, Subprocesses in decision making/Application, Knowledge type/Nature... 123
Table 6.2. Overview of biases (adapted from Hogarth, 1987) 134
Table 6.3. Summary of Evaluation Activities 163

TABLE OF CONTENTS

Executive Summary ix
0.1. Summary of the Study ix
0.1.1. Survey ix
0.1.2. Workshop ix
0.1.3. COADE Background x
0.1.4. COADE Concepts x
0.2. Main Conclusions xii
0.3. Major Recommendations xiii
0.4. Military Implications xiv
Chapter 1. Introduction 1
1.1. Background 1
1.2. Method of Work 2
1.3. Status of COADE 2
1.4. COADE Assumptions 3
1.5. How COADE is Expected to Work 3
1.6. Organisation of the Report 4
Chapter 2. Survey of C2 Decision Aids Development 5
2.1. Survey Responses 5
2.2. Follow-up Interviews on Cognitive Analysis Techniques 9
2.3. Summary of Findings from Decision Aids Survey and Interviews 10
Chapter 3. Summary of Workshop on Draft Framework 11
3.1. Goals and Assumptions of COADE 11
3.2. Cognitive Task Analysis 12
3.3. Supporting Analyses 12
3.4. Design 12
3.5. Evaluation 13
3.6. General Issues 13
Chapter 4. Introduction to the COADE Framework 15
4.1. Introduction 15
4.1.1. The Human in the System 15
4.1.2. System Development Steps 16
4.1.3. Iterative Development 17
4.2. Overview of COADE Activities 18
4.3. ANALYSE 18
4.3.1. Task and Performance Analysis 19
4.3.2. Cognitive Task and Cognitive Performance Analysis 19
4.4. DESIGN 20
4.4.1. Solution Approach Determination 20
4.4.2. Aiding Concept and Design Requirements Configuration 20
4.4.3. Human-Computer Interface 21
4.5. EVALUATE 21
4.5.1. Evaluation during ANALYSE 21
4.5.2. Evaluation during DESIGN 21
4.5.3. Summative Evaluation in EVALUATE 22
Chapter 5. COADE Framework 23
5.1. ANALYSE 26
5.1.1. Task Analysis 27
5.1.1.1. Decision Makers and Organisational Analysis 30
5.1.2. Performance Analysis 32
5.1.2.1. Performance Measures (R.1) 34
5.1.2.2. Performance Standards (R.2) 37
5.1.2.3. Types of Behavioural Errors (R.3) 39
5.1.3. Cognitive Task Analysis 43
5.1.3.1. Knowledge Elicitation (R.4) 45
5.1.4. Cognitive Performance Analysis 46
5.1.4.1. Cognitive Limitations (R.5) 48

5.2.1. Solution Approach 51
5.2.1.1. Selection and Design of Aiding Concept 55
5.2.1.2. Performance Aiding Strategies and Guidelines (R.6) 56
5.2.2. Human-Computer Interface Design Activity 63
5.2.2.1. Selected Design Principles, Guidelines, and Standards (R.7) 65
5.2.2.2. Design Methodologies and Tools (R.8) 68
5.2.2.3. Models and Architectures (R.9) 73
5.2.2.4. Information Exchange (R.10) 76
5.2.2.5. Promising Interaction Techniques and Technology (R.11) 77
5.3. EVALUATE 79
5.3.1. Evaluation 80
5.3.2. Evaluation Activities during ANALYSE 82
5.3.2.1. Verify Behavioural Model 83
5.3.2.2. Verify Cognitive Model 84
5.3.2.3. Verify Cognitive Requirements 85
5.3.2.4. Verify Performance Measures 87
5.3.3. Evaluation Activities during DESIGN 88
5.3.3.1. Verify Solution Concept 89
5.3.3.2. Assess Adequacy of Design 90
5.3.4. Summative Evaluation Activities 91
5.3.4.1. Assess Task and Organisational Performance 92
5.3.4.2. Evaluate Decision Aid Usability 93
5.3.4.3. Evaluate Technical Performance 94
Chapter 6. COADE Background 95
6.1. Models of Decision Making 97
6.1.1. Classical Models 97
6.1.2. Naturalistic Models 99
6.2. A General Cognitive Model of C2 Decision Making 103
6.2.1. Knowledge Representation & Memory 104
6.2.2. Monitor 107
6.2.3. Basic Processes 108
6.2.3.1. Understanding 108
6.2.3.2. Reasoning 108
6.2.4. Meta-cognition 109
6.2.5. Act 109
6.2.6. Learn 109
6.2.7. Situational Factors Influencing the C2 Decision Process 109
6.2.7.1. Familiarity - High 110
6.2.7.2. Familiarity - Moderate 110
6.2.7.3. Familiarity - Low 111
6.2.7.4. Effects of other factors 111
6.3. Cognitive Task Analysis 113
6.3.1. Introduction 113
6.3.2. What is Cognitive Task Analysis? 113
6.3.3. Cognitive Model Generation 115
6.3.4. Knowledge Elicitation 115
6.3.4.1. Approaches to KE 116
6.3.4.2. Declarative Knowledge 116
6.3.4.3. Procedural Knowledge 116
6.3.4.4. Strategic Knowledge 117
6.3.4.5. KE Techniques 117
6.4. Cognitive Performance Analysis 131
6.4.1. Implications of Cognitive Task Analysis for Performance Analysis 131
6.4.2. Approaches to Cognitive Performance Analysis 132
6.4.2.1. Structural Constraints 132
6.4.2.2. Cognitive Workload 133
6.4.2.3. Comparison to Ideal 134
6.4.2.4. Amount of Deliberation 135
6.4.2.5. Case-based Findings 136

6.4.3. Cognitive Limitations 136
6.4.3.1. Schema Limitations 137
6.4.3.2. Learning Errors 138
6.4.3.3. Errors in Knowledge Representation 138
6.4.3.4. Basic Processing Errors 138
6.4.3.5. Limitations of Meta-cognition 139
6.4.4. Analysis of Cognitive Limitations 140
6.5. Performance Aiding Strategies and Guidelines 143
6.5.1. Aiding Approaches 143
6.5.2. COADE Performance Aiding Strategies 144
6.5.2.1. Control of Reasoning Processes (Meta-cognition) 145
6.5.2.2. Reasoning Processes (Cognition) 147
6.5.2.3. Information Handling 148
6.5.3. Guidelines for Designing Decision Aids 149
6.5.3.1. Guidelines Derived from Failed Decision Aids 150
6.6. Human-Computer Interface Design 153
6.6.1. The Concept of Usability 154
6.6.2. Selected Design Principles, Guidelines, and Standards 155
6.6.3. Design Methodologies and Tools 155
6.6.3.1. Design Methodology 155
6.6.3.2. User Interface Design Tools 156
6.6.3.3. Rapid Prototyping 156
6.6.4. Models and Architectures 157
6.6.4.1. The Use of Models 157
6.6.4.2. Dialogue and Interaction Architecture 157
6.6.4.3. Information Exchange 158
6.6.5. Promising Interaction Techniques and Technology 159
6.6.5.1. Intelligent Interfaces 159
6.6.5.2. Multimodal and Multimedial Dialogue Design 159
6.6.6. Dialogue Summary 159
6.7. Evaluation 161
6.7.1. Introduction 161
6.7.2. Framework for Decision Aid Evaluation 161
6.7.2.1. Evaluation Factors 161
6.7.2.2. Relationship between Evaluation and Other Activities 162
6.7.2.3. Issues in Decision Aid Evaluation 163
References 165
Annex I Terms of Reference RSG.19 183
Annex II RSG members and participants 185
Annex III Workshop participants 187
Annex IV TOR Ad Hoc Group 189
Appendix A. Concepts used in Modelling Cognition 191
A.1 Introduction 191
A.1.1. Overview of Approaches to Cognitive Modelling 193
A.1.2. Fundamental Characteristics of Cognitive Entities 193
A.2. Elements of the Cognitive System 195
A.2.1. Memory 195
A.2.2. Representation of Knowledge 196
A.2.2.1. Types and Organisation of Knowledge 197
A.2.2.2. Knowledge on Concepts and Relations 197
A.2.2.3. Knowledge of Events and Situations 198
A.2.2.4. Hybrid Representations and Mental Models 199
A.2.2.5. Summary 200
A.2.3. Basic Processes and Operations in Cognition 200
A.2.4. Meta-cognition 202
A.2.5. Learning 203
A.3. Schema-Based Problem Solving 204
A.4. Implications for COADE 206

This page has been left blank intentionally

EXECUTIVE SUMMARY

0.1. SUMMARY OF THE STUDY

i. The NATO Research Study Group on Decision Aids in Command and Control (RSG.19) was tasked by Panel 8 of the NATO Defence Research Group to develop a framework for decision aid design and evaluation. The background of this tasking was the conclusion by RSG.12 that most decision aids fail to provide adequate support because of a lack of knowledge concerning decision making activities in command and control (C2). RSG.19 reformulated the problem of decision support as a more general system design problem:

a) how to identify critical cognitive requirements?
b) how to translate those into system design decisions?
c) how to evaluate the adequacy of the solution?

RSG.19 has developed COADE (Cognitive Analysis, Design, and Evaluation), a framework that addresses the cognitive human factor in systems design as an approach for considering these issues.

ii. RSG.19 gathered material from the literature, and developed new concepts when these were lacking. Additionally, RSG.19 performed a survey on the use of cognitive techniques in system development, and organised a workshop to expand earlier COADE concepts.

iii. COADE provides the developer, and in particular the cognitive specialist in the development team, with an approach for development of cognitively-centred systems, an overview of methods and techniques, summaries from reference material, guidelines for aiding, and an overview of literature related to C2 and decision making processes. In addition, a cognitive description language that can be used to describe tasks in a standard way is proposed.

0.1.1. Survey

iv. The goal of the survey was to gather information from developers with C2 decision aid experience. Of 166 surveys sent out, there were 26 usable responses. Only one response related to a deployed tactical aid. Ten respondents had some experience with cognitive analysis. Five follow-up interviews were performed. The opinion of the respondents was that users should be responsible for defining requirements. Typically, requirements are stated in general and indefinite terms ('better', 'faster'); specification in quantitative performance terms is rare. There were no specific methods suggested for analysing cognitive aspects of decision making and interpreting aiding requirements. Evaluations were informal, often no more than demonstrations with subjective feedback. Effectiveness of performance was the most frequent evaluation criteria. Three-quarters of the respondents indicated that existing methods for decision aid development are inadequate and that the methods should incorporate cognitive task analyses. Three-quarters of the respondents would use methods that better integrate cognitive capabilities, if available. The survey and interviews indicated that current development practices for C2 decision aids are inadequate and that they can be improved by integrating techniques to deal with human factors and cognition.

0.1.2. Workshop

v. The goal of the workshop was to evaluate a first draft of the COADE framework with experts from the field of decision aid development. The seven invited experts delivered position papers concerning decision aid development with reference to COADE. The value of having a framework such as COADE was underlined by all experts. Establishing methods and techniques for cognitive analysis was seen as particularly critical, although the experts themselves were content with their own strategies and individual mixtures of methods. They felt that performance standards are important for identifying deficiencies and targeting design solutions. Cognitive requirements should be embedded in the technical requirements. The analysis and rationale on which the requirements specification is based should be given to designers so that they understand what is needed. The benefit of cognitive analysis depends on the involvement of the cognitive analyst at every step in the development process. The participants agreed that evaluation should be an ongoing iterative process.

The development of scenarios and performance measurement techniques are critical aspects of evaluation. A representative range of scenarios should be used in evaluation. The experts suggested making the COADE framework more 'alive' and including examples of application of the framework. Use of pictures and tables would make the document more accessible. Finally, they advised RSG.19 to develop COADE in an evolutionary way.

0.1.3. COADE Background

vi. A system is defined as a set of interdependent elements, human and/or machine, that produces certain behaviour. C2 systems vary in the mix of human and machine elements and in the roles ascribed to each. At higher levels of C2, e.g. Corps level, the systems consist of mainly human elements; at lower levels, machines such as tanks, are essential elements of the system. The dramatic increase in the use of computers is changing the role of the human: at lower levels machines are integrated to replace the human, at higher levels computers support the human decision maker. Although the focus of RSG.19 is on decision support, RSG.19 advocates a human-centred approach to system change and development on all levels. Even if a change is purely technical, such as an increase of information transmission speed, the consequences for human performance should be assessed. For example, one consequence of increasing information speed might be a requirement for faster human information processing and decision making.

vii. A human-centred approach takes the view that the human has the central role in a system and is ultimately responsible for achieving the system goals. COADE addresses the cognitive processes involved in complex task situations. The central theme in a cognitive approach is that an understanding of the mental processes, goals, and knowledge is necessary since cognition is the base of decision making.

viii. The goal of COADE is to assist in the development of decision support in complex systems, based upon knowledge of the human's role, capabilities, and the tasks to be performed. It aims at reducing the risk of addressing the wrong problem, or choosing the wrong solution. COADE does not assume a strict procedural order of activities, although it advocates proper identification of the support problem before solutions are developed. COADE provides a method for contributing knowledge concerning human needs and limits to all phases of system development. COADE activities should be started in the early phases of the development, such as system concept definition and system requirements analysis.

ix. It is assumed that systems analysis and development will be performed by a team of specialists, one of whom addresses the behavioural and cognitive aspects of the system. Because of the unique character of cognitive analyses it is assumed that this specialist will have a background in behavioural or cognitive science. This (cognitive) specialist will be the main user of COADE.

0.1.4. COADE Concepts

x. COADE recognises three main sets of activities in decision support development: ANALYSE, DESIGN, EVALUATE (see Figure 0.1). Each set of activities views the decision support problem from a different perspective:

a) In ANALYSE, the goal is to identify the critical aspects of the system; the <u>what</u> of the decision problem. Analysis results in a specification of critical cognitive requirements on which design decisions should be based.
b) In DESIGN, the goal is to find a solution for the identified problem; the <u>how</u> of the solution to the problem. One possible solution is the use of computers to support the human cognitive performance as a means of improving the functioning of the system.
c) In EVALUATE, the developer steps aside for a moment and verifies the results of the activities in a more or less formal way. Evaluation is done continuously on the results and products of analysis and design in order to guarantee or control the quality of the activities performed. A final evaluation tests the

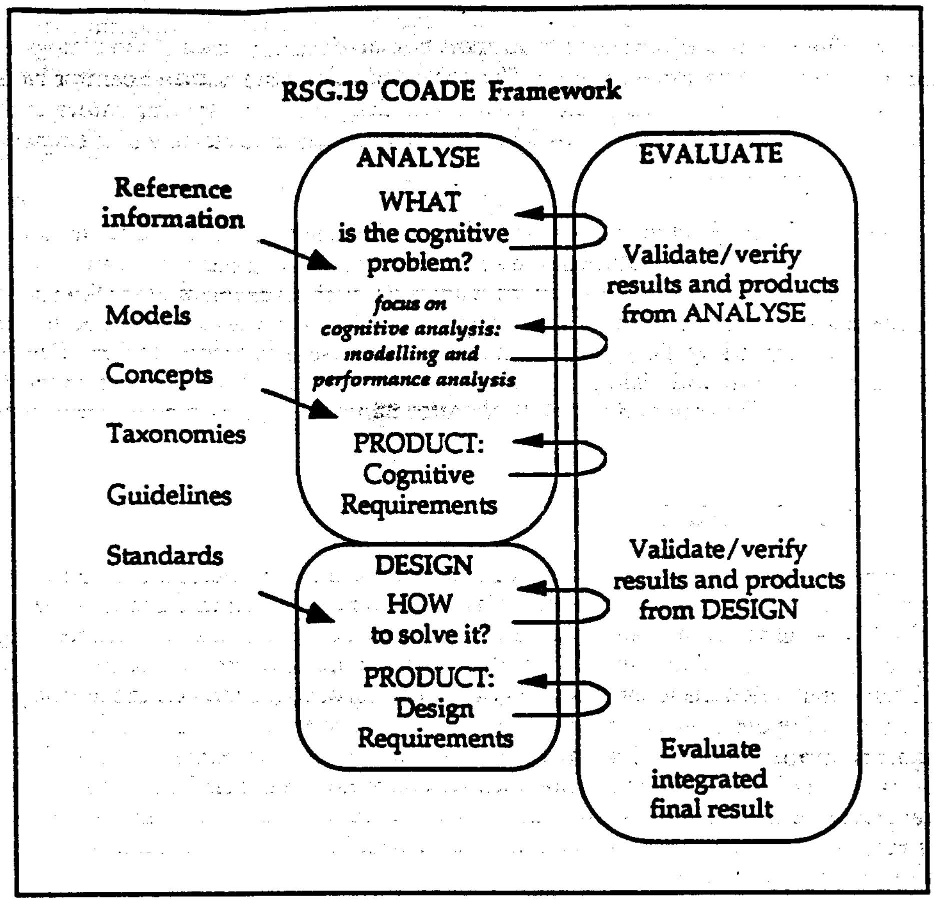

Figure 0.1. Main Components of the COADE Framework: ANALYSE, DESIGN, and EVALUATE Activities with Reference Information.

xi. ANALYSE comprises the most extensive set of activities. Also included are behavioural types of analyses, such as the traditional task analysis, which permits the positioning of the cognitive analysis in the broader context of system performance. The development of a cognitive model of the human activities in the system is a central activity in COADE, because it provides the basis for a determination of how well the human performs cognitively. The COADE framework provides an overview of knowledge elicitation techniques that help capture the knowledge used in cognitive tasks. The cognitive performance analysis results in a specification of cognitive requirements, critical aspects of the human performance that must be addressed in the design solutions. COADE provides an overview of cognitive limitations and biases, and human error taxonomies.

xii. The main difficulty in describing cognitive behaviour has been the lack of a generic set of concepts (a cognitive descriptive language) that captures the crucial aspects of human cognitive processes in realistic tasks. COADE proposes such a set of concepts in a general model of Command and Control decision making. Central is the concept of the mental schema in which are linked the general and specific facts, goals, and actions concerning particular situations in the world.

xiii. In the background sections accompanying COADE, current theories of decision making in realistic task situations are reviewed. Aspects of C2 that are crucial factors in human performance are also discussed. These discussions provide information that can be used to focus the analysis efforts on the relevant aspects of the system.

xiv. DESIGN concerns the translation of decision support requirements into solutions. COADE stresses the consideration of different perspectives on solutions during trade-off analyses. COADE provides a taxonomy of performance aiding strategies and guidelines for designing decision aids. The aiding strategies address three clusters of human cognitive processes: meta-cognition, cognition, and information management

xv. When computers are used to support human decision making, the dialogue between human and computer becomes an issue. The goals and role of the human operator have been specified in the analysis activities. At this point, a further detailing out of the specific nature of the human-computer interaction is required. COADE therefore provides an overview of dialogue design issues and methodologies and tools.

xvi. EVALUATE activities vary according to the concreteness of the solution to the problem. The more concrete the specification of the solution is, the more precise the evaluation can be. COADE stresses the formal evaluation of results from analysis, such as problem identification; behavioural and cognitive models; crucial cognitive limitations. Evaluation activities are intended to verify that these conclusions are shared by the experts from the user community, or are confirmed by simulations or experiments. Solutions and dialogue concepts proposed during design should be formally evaluated by measuring the efficiency and accuracy of performance using experimental prototypes.

0.2. MAIN CONCLUSIONS

xvii. Decision aids in C2 applications have not been very successful as indicated by the low ratio of fielded aids to aiding attempts. This observation comes from personal experiences of RSG.19 members, a panel of seven subject matter experts, and a survey of decision aid developers and evaluators. The COADE framework was developed to improve the success rate of decision aids for C2. This framework aims to incorporate critical aspects of cognition into the development process and to make resulting decision aids truly supportive of decision makers. COADE incorporates analysis, design, and evaluation activities that should be followed in development. COADE stresses the analysis activity because too many previous aids have not been based on a good understanding of actual performance and the cognition underlying decisions. Earlier system requirements have been too strongly founded on shallow understandings of C2 tasks, on what technologies are available, or on user desires rather than needs.

xviii. The original plans of RSG.19 were to establish human factors principles and methods for evaluating C2 decision aids. From the beginning of the Study, it became evident that evaluation methods alone would not be sufficient for making aids more responsive to human decision making needs. Methods needed to be identified that could show how to understand the influence of human factors and to develop the requirements for what could be done to aid them. RSG.19 also needed to distinguish the prominence of the decision making aspects of C2 situations from the sensory and motor processes which are typically associated with human factors. Thus, RSG.19 focused on cognition and developed a cognitively-centred framework for analysis, design, and evaluation.

xix. There were several areas that initially were assumed to be established well-enough to incorporate directly into a new cognitive development methodology. However, RSG.19 discovered that existing cognitive task analysis methods did not provide sufficient guidance. Also the body of knowledge about cognition was not well integrated nor directly transferable to improved development procedures. Because of this shortfall, the Study Group compiled a set of baseline cognitive concepts and a general model of C2 decision making. These concepts and model are an important part of the analyse and design activities. A hybrid method was developed: the cognitive concepts and model of C2 decision making provide a basis for the development of a model of cognitive performance and, subsequently, this model is input for a critical analysis of cognitive limitations.

xx. Another innovation that RSG.19 used was to differentiate between descriptive task analysis and analytical performance analysis. Task analysis is an important supporting method for better understanding the task and situational environment in which decisions are made. However, most task analyses are descriptive in nature indicating how a task is performed, but they are not evaluative and do not indicate how well a task can be performed. The Study Group initiated the concept of performance analysis for assessing how well tasks are performed. A distinction was also made to show that task analysis primarily focuses on observable behaviours, while cognitive task analysis addresses the underlying cognitive processes and knowledge for those behaviours. The concept of performance analysis was extended to cognitive analysis to infer the types of cognitive limitations that

xxi. One of the major characteristics of complex C2 decisions is that context is critical. RSG.19 investigated new models of decision making and reviews them in this report. The field of decision making seems to be on the verge of a major paradigm shift from classical models to natural, practical, everyday ones. The new models, often referred to as "naturalistic", contrast with classical models that provide a prescription of optimal decisions. The new models try to explain actual behaviour instead of an ideal originating from a view that humans should display "rationality" as defined by economic theory. Naturalistic theories put greater emphasis on the actual characteristics of decision makers and the situations in which they act, as well as taking a broader perspective on what are the important tasks. RSG.19 was sensitive to this new direction and included background information in COADE that provides characteristics of the decision makers and situational dependencies for decision making.

xxii. Besides advocating a cognitively-centred approach that puts greater emphasis on specific characteristics of decision makers, tasks, and the situation, the COADE framework also recommends that a computer-based aiding solution not be assumed to be the only correct approach. Other solution approaches need to be considered based on the problem, the identified cognitive requirements, and the feasibility of meeting the requirements with different solutions. Aiding solutions come to mind first, but other solution approaches like training, procedural change, or personnel change may be more appropriate. If an aiding solution is warranted, an analysis should be conducted to predict what changes the aid will have on the rest of the system. It is likely that different training, personnel, or procedures will be needed with the incorporation of an aid.

xxiii. Decision aids do not simply allow for faster, more accurate processing of information, they can change the nature of the task for the decision maker. These considerations are brought out further in a taxonomy of decision aiding strategies that the framework offers. The taxonomy groups candidate strategies into cognitive, meta-cognitive, and information handling concepts. The taxonomy serves as a guide to satisfy cognitive requirements that are determined in the analysis stage of the process. Although a "deductive" procedure was initially sought for matching requirements with aiding strategies, it was soon realised that such an approach would be ill-fated. Knowledge about cognitive limitations and remediation is not so definite and intractable that a matching technique would be practical or desirable.

xxiv. Throughout the proceedings of RSG.19 it was detected that decision aid development is rarely initiated by an intentional desire to better understand the decision problem or see if an aid is warranted. There is a real danger of failing to develop useful aids if developers respond to requirements that have not been validated through careful analysis and evaluation. Technology-based developments may seem desirable to many, but the resulting aids are very likely to overlook real underlying problems of cognition. Only when the problem is understood and validated and solutions are derived accordingly will there be a sound basis for developing an aid.

xxv. COADE provides a cognitively-oriented approach that aspires to get development efforts better focused on real decision making concerns, that address the underlying cognitive aspects of performance. If we want decision aids that improve decision making, it makes sense that the emphasis be placed on making aids that are responsive to cognitive requirements, rather than ones that simply advocate technology or replace human performance with automation of selected tasks. RSG.19 believes that COADE places the spotlight on the essence of decisions -- the decision maker's cognitive capabilities and limits -- and that COADE can beneficially influence the development process to produce decision aids that improve overall performance capabilities.

0.3. MAJOR RECOMMENDATIONS

xxvi. The research field from which COADE derives its approach is young and under development. The methods are not yet well-established. The use of cognitive analysis requires thorough training in behavioural and cognitive science. There exists almost no systematic methodology for the application of cognitive factors in system design. COADE proposes such a methodology. Although this document is the final report of RSG.19, given the state of the field, an evolutionary approach to upgrading COADE would be highly opportune.

xxvii. Panel 8 should support the evolutionary development of COADE. COADE is not merely

a collation of information already available on the development of decision support. Rather, it is a new approach to cognitively-centred systems. COADE itself should be seen as a prototype that requires at least another evaluation cycle. It is proposed that a follow-up Ad Hoc Group (see Annex IV) evaluate COADE by means of

a) peer reviews
b) application of the framework in current or proposed system developments.

xxviii. Panel 8 should support the development of a hybrid methodology in which cognitively-centred engineering and systems engineering are integrated in a human-centred system design methodology. The work of RSG.14 and RSG.19 provides a basis for such a methodology. This methodology could build upon military standards for human engineering requirements (US MIL-H-46855B) and system software development (DOD-STD-2167A). In addition, there is a need to develop a management structure for Project Officers to support the Project Management Office and Procurement in the specification of cognitive systems engineering and the management of command and control information system (CCIS) development.

xxix. Panel 8 should support the monitoring of cognitive systems engineering efforts in CCIS developments in the NATO countries. The monitoring should result in a description of the problems identified, the solutions chosen, and, in particular, how cognitive engineering is applied in the development. The group of experts that does the monitoring could advise the development teams for new CCISs on how to apply cognitive systems engineering.

xxx. COADE should be used in host nation and NATO CCIS and decision aid developments and other initiatives with higher order cognitive components.

0.4. MILITARY IMPLICATIONS

xxxi. The introduction of computers into military forces provides an opportunity for support of human cognitive performance in planning and decision making. This opportunity should be exploited in CCIS development along with the information handling applications of the computer, such as faster switching, storing, and processing. Any system development plan should contain a section on support for planning and decision making.

xxxii. Instead of waiting for contractors to apply human factors and cognitive systems engineering, the military should specify what is required in this regard. Military standards that address the cognitive aspects of human factors requirements and methods should be developed and their use be made mandatory in system development.

xxxiii. More attention should be devoted to human factors and cognitive systems engineering in the training of project officers. Besides systems engineering approaches, human-centred and cognitively-centred approaches should be presented in order to increase awareness of human factors and cognitive issues. The training should highlight what happens when cognitive factors are ignored and what the benefits are when they are considered.

xxxiv. In the early conceptual phases of CCIS procurement and development the military should consult specialists in human factors and cognitive factors in order to

a) identify what can be improved in the existing system
b) determine the opportunities for cognitive support in projected systems
c) assess the consequences of these factors in the definition of future systems.

CHAPTER 1. INTRODUCTION

1.1. Background
1.2. Method of Work
1.3. Status of COADE
1.4. COADE Assumptions
1.5. How COADE is Expected to Work
1.6. Organisation of the Report

1.1. BACKGROUND

1. A standard opening line for textbooks dealing with the role of the human in man-machine systems asserts that the increased use of computers has caused the role of the human to become more cognitive. From being formerly psychomotor human-machine interactions, human tasks have shifted towards supervisory and decision making tasks. In the area of C2, as well, the computer was introduced to provide machine-like assistance, such as more rapid switching, storing and processing of larger amounts of information. Attention is now shifting to the use of computers to support human cognitive performance, with the intention of improving overall system performance. COADE aims to support human and cognitively-centred system development by

a) helping to identify the cognitive constraints in actual and future task situations;
b) providing aiding concepts to address those constraints;
c) organising the development in terms of generation and evaluation and evaluation in terms of modelling steps of these products.

2. One of the concluding findings of the NATO Research Study Group-12 on Computer-Human Interaction in Command and Control was that focused research is needed on the development and integration of computer-based decision aids for command and control systems (McCann, 1989). Therefore, the NATO Research Study Group-19 on Decision Aids in Command and Control (RSG.19) was tasked by Panel 8 of the NATO Defence Research Group to develop a framework for decision aid design and evaluation. RSG.19 first met in the spring of 1990 to begin this work and had its eighth and last meeting in fall 1993.

3. A sensitivity to cognitive factors is a prominent characteristic of the work under Panel 8. The need for research into cognitive factors had been expressed previously in a workshop on "Applications of system ergonomics to weapon system development" (Merriman, Muckler, Howells, Olive & Beevis, 1985). RSG.14, another body under Panel 8, addressed the effectiveness and use human engineering techniques in system development. They recommended that the Panel support research and development of function allocation and task analysis techniques dealing with cognitive behaviour (Beevis, 1992). We believe that the framework developed by RSG.19 fills the gap identified by these bodies, in that it addresses the cognitive factors in system development.

4. The Terms of Reference of RSG.19 (see Annex D) state three main objectives:

a) bring together existing knowledge on practical human decision making in Command and Control
b) assess, from human factors viewpoint, existing and proposed decision aids
c) develop framework for the human factors evaluation of decision aids.

In the course of the RSG discussions it became clear that instead of assessing the value of decision aids after the fact, a more pro-active approach was needed. The development of decision aids should be based on requirements that result from an in-depth and detailed analysis of the cognitive problems in the task situation under analysis. Consequently, the focus was changed towards identifying a cognitive approach to system development. We found that no comprehensive approaches existed. The challenge was to come up with an approach ourselves, incorporating what was available and adding what was thought to be essential for cognitively-centred system development.

5. The long term objective of the COADE framework is to improve command and control

decision making by basing decision aid development on cognitive requirements of the decision makers. The goal of COADE was defined as follows:

> "To assist in the design, test, and evaluation of computer-based decision aids, based upon knowledge of the human's role, capabilities, and tasks to be performed in command and control."

6. COADE is not written as a cookbook for system developers with a step-wise procedure for applying cognitive analysis. The user is expected to be a behavioural specialist who knows about human factors issues and is a member of a development team. COADE provides this specialist with a list of relevant activities, references and a global structure for cognitive analysis. The developer can use COADE to see what could be involved in the application of a cognitively-centred approach, or can use COADE to require particular activities from contracted specialists. He or she will find in COADE

a) an approach to cognitively-centred specification of system requirements;
b) a set of concepts and a general model of C2 decision making that can be used to describe functions and tasks;
c) a discussion of decision making task situations and Command and Control characteristics;
d) a compendium of methods and techniques for the description and analysis of cognitive performance;
e) an overview of known limitations and errors in cognitive performance;
f) guidance in the translation of cognitive requirements into design requirements;
g) an overview of issues in the design of user interfaces;
h) an evaluation approach to improving the quality of the intermediate results and products.

1.2. METHOD OF WORK

7. The members of RSG.19 (see Annex II) conducted the work through correspondence (e-mail), meetings twice a year, an additional subgroup meeting, and a workshop. Early on in the project, a survey of developers concerning cognitive analysis as applied in the development of decision aids was carried out. About mid-way through, a workshop with experts in decision aid development was held. The project itself was divided into two phases: The first phase of work was to specify the analysis, design, and evaluation activities that should be included in the COADE framework. This phase was closed off with a discussion of the framework as it had been developed to that point at the workshop (1992). This provided input for the second phase of the development. In the second phase, attention was given to the development of background material, in particular, of cognitive concepts that can be used in the analysis of cognitive task situations. Annual reports and verbal presentations informed Panel 8 about the progress and the results of the work. The final report closes off the work of the RSG.

1.3. STATUS OF COADE

8. A framework organises the knowledge that is available about an area into an integrated structure. Frameworks are used to organise areas that are new, incomplete, or a-theoretical. The field of cognitive analysis is young and actively under development; its methods are not yet well-established; and it demands an analyst with thorough training in behavioural (cognitive) science. Moreover, there is little systematic methodology for the application of cognitive factors in system design. COADE specifies activities and methods to systematically assess the cognitive aspects of the tasks in the system, and gives guidance in finding design solutions based on these assessments.

9. The current version of COADE has two main parts. One part contains the actual framework with a description of activities, products, and methods. The other part contains background information concerning the activities, and the cognitive concepts involved in general and in C2 decision processes. An earlier version of the actual framework was reviewed by experts at a workshop and extended as a result of the discussions.

10. Although COADE is a rich source for cognitive concepts and techniques, no claim of exhaustiveness is made. Given the status of the field of cognitively-centred system design, COADE

cannot be a finished product; an evolutionary approach to updating COADE is highly opportune. The intention is to augment the current version through two activities:

a) solicit comments on the current version (framework and background material) from experts in the field;
b) apply COADE in concrete projects, report on these applications, and modify and update COADE accordingly.

1.4. COADE ASSUMPTIONS

11. A first set of assumptions relates to the how to cure the problem of inadequate decision support systems. It is assumed that the problem is twofold. It can be due to either a failure to understand the human aspects of the functions and tasks in the system, or a failure to select or generate an adequate solution to the cognitive problem. A better understanding of the cognitive factors will result in a system that is better tuned to the human which, consequently, will improve performance of the system as a whole. A thorough analysis of the role of the human, the task situation, and the capabilities of the human is one cure which will result in better specifications of the cognitive requirements of the system. Analysis will reduce the risk of trying to solve the wrong cognitive problem.

12. The second aspect of the problem is finding an adequate or good-enough solution. This is partly dependent on the creativity and experience of the developer. The use of design methods that include evaluation activities will reduce the risk of making wrong or impractical choices. Analysis, design, and evaluation are distinctive, but closely related activities performed in an iterative fashion. Hypotheses formed during analysis and design are tested in the evaluation activity. Results from evaluation and design feed further analysis.

13. A second set of assumptions relates to the approach used for system development. COADE is not confined to one particular system life cycle model, but it will work best with models that stress the user's requirements as a basis for system requirements. COADE defines a system as comprising humans and machines working to achieve objectives; the central premise is that systems ultimately exist for human purposes. This, then, demands a human-centred approach to system development. Starting from the goals (missions) that the system wants to achieve, the question is the structure and timing of the required overt user behaviour; thus, a behavioural analysis provides the backbone for the cognitively-centred analysis. The cognitive analysis provides a model of the mental processes that are involved in the task behaviour. Therefore, the development of a behavioural model and a cognitive model are crucial activities in the development process.

14. Analysis activities define system requirements and should therefore precede design activities in which solutions are developed. Design activities define more detailed requirements that are linked to the particular solutions chosen. Evaluation activities (verification and validation) should be regarded as providing quality assurance for the results that are produced during the development process and not solely as a last step in the development.

1.5. HOW COADE IS EXPECTED TO WORK

15. It seems that designers use information from research in only a discretionary fashion (Rouse & Boff, 1987a). Part of the problem is probably that this information is not in an easy to use form. An increase in the use of information by developers might be achieved by making it more available (handbooks), and more accessible (computer-based guidelines). A more basic problem, however, may be the attitudes held by developers. If a developer does not believe that proper handling of human behavioural and cognitive issues will make a major difference in the final acceptance of the system, he or she will like not bother to incorporate behavioural and cognitive analyses in the management plan. The attitude often is "What guarantee can be given that investment of effort in these issues will reduce the risk of the project and result in a better product (i.e., what is the 'business case')?" In COADE, no special effort will be made to convince developers of the importance of addressing human factors. We assume that the developer is committed to basing the specification of system requirements on behavioural and cognitive analysis, standards and guidelines, either because of contractual obligation, competitive edge, or common sense. COADE comprises activities that

produce validated system requirements based on cognitive characteristics of human, task, and environment.

16. The analysis of human behaviour is a specialised activity that requires specific expertise. We assume that a behavioural specialist or person with equivalent training is a member of the development team. COADE does not provide the fundamentals of cognitive engineering, although some effort has been put into providing background information concerning concepts and methods used in cognitive analysis.

17. COADE provides input to all phases of system development, such as system concept definition and system requirements analysis (Department of Defense, 1988; Overmyer, 1990). COADE activities should preferably begin in the early phases of the development, because a focus on the critical issues of the system which will reduce the loss of time in trying to implement the wrong solution. the gain that can be achieved by early focus on real problems. When the system procurement strategy involves procurement of 'commercial-of-the-shelf' components, rather than an in-house development, COADE supports the specification of requirements and criteria for the assessment of the adequacy of ready-made solutions.

18. The COADE framework guides the specialist or analyst and the development team in the application of cognitive analysis, design, and evaluation 'activities'. An 'activity' is a specific partitioning of effort suggested within analysis, design or evaluation. The specialist can use the set of activities to decide which are to be performed in the project. The COADE framework provides a general description of each activity, a specification of products, and methods, and relationship to other activities. The reference information sections provide a short overview of the literature related to the activity. The specialist can use a reference information section to see what the relevant concepts and literature are. In addition, extended discussion and background information concerning the activities and the cognitive issues and concepts involved is provided in a separate chapter.

1.6. ORGANISATION OF THE REPORT

19. Chapters 2 and 3 contain the results of the SURVEY and the WORKSHOP, respectively. Chapter 4 gives an overview of the COADE activities. Chapter 5 contains the actual COADE framework organised in three sections:

ANALYSE
DESIGN
EVALUATE.

20. Chapter 6 and Appendix A contain the sections on specific issues and concepts relevant for the application of COADE. The following issues are covered:

a) Cognitive modelling, in particular, the concepts relevant for the description of cognitive tasks, such as a schema-based model of problem solving and C2 decision making;
b) Models of (C2) decision making and factors that influence the C2 decision process;
c) An overview of cognitive analysis and knowledge elicitation techniques and their strengths and limitations;
d) A framework for cognitive performance analysis, including an overview of biases, cognitive limitations, and human error classifications schemes;
e) A taxonomy of performance aiding strategies is proposed to support the transition from cognitive requirements to potential solutions and design requirements;
f) Issues in human-computer interface design with reference to design principles, guidelines, and standards and methodologies and tools;
g) Issues in evaluation in relation to the analysis and design activities.

CHAPTER 2. SURVEY OF C2 DECISION AIDS DEVELOPMENT

2.1. Survey Responses
2.2. Follow-up Interviews on Cognitive Analysis Techniques
2.3. Summary of Findings from Decision Aids Survey and Interviews

21. A survey was carried out by RSG.19 to gather information from individuals with C2 decision aid experience. The survey was targeted at developers of decision aids. The names and addresses were assembled from those proposed by RSG.19 members and from a list of members in a US decision aiding working group. Follow-up, telephone interviews were conducted with five respondents who were among those reporting having done a cognitive analysis. The interpretation of their comments follow the survey findings.

22. The survey addressed five topics. Background information was requested on the individual's role in the development, the organisation, and descriptive information about the aid. Questions on the requirements of the aid addressed what initiated the development and whether the aid was meant to address specific human limitations. Questions on the development asked specifically about the methods followed, including rapid prototyping, task analysis, cognitive analysis, and the selection of aiding concepts. Questions on evaluation covered items like whether there was a formal evaluation, what means were used to collect the data, and whether a cognitive analysis influenced the evaluation. General questions asked for comments on the adequacy and suggested changes in the development process, whether cognitive analysis methods would be used (if available), and ideas for how to accomplish different development processes within cost and schedule constraints.

23. Of 166 surveys sent, there were 26 usable responses (16 percent return rate). The 26 respondents included 15 developers of tactical prototypes, 5 developers of non-tactical aids, 2 evaluators, and 4 other. Only one response related to a deployed tactical aid.

2.1. SURVEY RESPONSES

24. Table 2.1 gives selected questions from the survey. Following the questions (Q) are the categories of responses and frequencies.

Table 2.1. Responses from C2 Decision Aid Development Survey

Q1. Technologies used in decision aid?	
Expert system	9
Multi-attribute utility approach (MAUA)	4
Simulation, analysis	4
Algorithmic, computational	3
Object oriented model	3
Knowledge based	2
Information retrieval	2
Geographic information analysis	1
Matrix manipulation	1
Artificial intelligence planning	1
Loosely coupled neural networks	1
Q2. Respondent's role in decision aid development?	
All aspects	14
Development	4
Requirements analysis	3

Evaluation	2
Knowledge engineering	1

Q3. What prompted the start of the decision aid development?

Government or user representative	14
Previous aid	3
Self (research, recognition)	3
Technology (explore)	2
Demonstrate method (development)	1

Q4. Was the original desire for the aid stated in quantifiable terms?

Yes	2
No	23

Q5. Was the aid meant to address specific human limitations?

General human limitations	28
Specific human limitations	6
Adhere to prescribed model	3

Q6. Do you have a specific method for C2 decision aid development?

Yes	4
No	9
Used general development methods	3

Q7. Was rapid prototyping used?

Yes	20
No	4

Q8. How many prototypes were there; over what time period?

Period	6 mo	1 y	1.5 y	2 y	3 y	4 y	5 y	6 y	8 y
Number	1			2					
	2		2	4	4	3-4			
	3	12	3-4	12	*	*	*	2	20

* continual prototypes

Q9. Were there any problems incorporating feedback?

Yes	7
No	11

Problems identified:

Conflicts in opinion	3
Insufficient funding	1
Insufficient time	1
Comments appropriate to fielded system not to requirement concepts	1

Q10. How was the decision making situation met?

Mentioned how information was obtained	13
Mentioned how information was represented	3
Mentioned analysis technique	2

Q11. How were areas to be aided selected?

Requirements were already provided	13
Based on available technology	6
Problem studied	4

Q12. How formal was the evaluation of the aid?

Informal, demonstration	12
Formal, performance testing	9
Both	3

Q13. Was the evaluation from the user perspective, system perspective, or both?	
User	9
System	5
Both	10

Q14. Were real scenarios used?	
Yes	22
No	4

Q15. Were real users used?	
Yes	20
No	4

Q16. Was there an evaluation independent of the developer?	
Yes	14
No	10

Q17. What were the primary issues in the evaluation?	
Performance, effectiveness	11
Ease of use	6
Speed	6
Accuracy	5
Knowledge base	3
Displays, interface	3
Safety	1
Viability, acceptability, validity	1
Confidence	1
Reasoning	1

Q18. Were there multiple evaluations?	
Yes	15
No	9

Q19. How was the evaluation reported?	
Formal reports	16
Informal reports	8

Q20. Was there an explicit relationship between the cognitive analysis (if performed) and the evaluation?	
Yes	6
No	2

Q21. Do you feel existing methods for C2 decision aids development are adequate?	
Yes	4
No	13

Q22. What's missing in the current process?	
Getting operational requirements	3
C2 & decision aiding theory and model	2
Method for cognitive requirements	2
Better knowledge elicitation	2
Team decision making	2
Iterations	1
Users not willing to consider new solutions	.1
Evaluation:	
Formal test & evaluation	1
Getting users	1
Experimental manipulation, measurement	1
Sufficiency of aid	1

Q23. What needs changing in current process?

Cognitive task analysis (functional)	3
Needs analysis, human factors analysis	2
Iteration, incremental builds	2
Patience with front end analysis	2
Evaluation, formal feedback	2
Appropriate time, funds, and management	1
Prototyping	1
Embedded AI tools	1
Better knowledge elicitation	1
Interaction between user-developer, users	1

Q24. What needs to be deleted in current process?

Detailed task analysis, front end analysis	2
Formal documentation	2
Early emphasis on building "something"	1
Full statement of requirements	1
Less emphasis on technical software issues	1
Rapid prototyping	1
Cognitive analysis	1

Q25. What were your key successes in developing C2 decision aids?

Improved development methods	10
The resulting aid	7
Advance technology	4

Q26. Would you use a different method (if available) to better integrate cognitive capabilities into the aid?

Yes	15
No	5

Q27. Changes in the development process must pay for themselves in increased performance within cost and schedule constraints. Please make any specific suggestions for how to accomplish this.

Research your topics to be modelled. Do a thorough needs analysis of what is really wanted now and in the future.

Cognitive task analysis provides better information and takes less time and effort than detailed task analyses.

More systematic use of cognitive requirements analysis and evaluation procedures introduced early in the development cycle.

Move cognitive task analysis to the centre and derive the task analysis from it. Consciously decide whether or not to build a decision aid instead of plunging into how to build one.

Need meaningful effectiveness metrics for aids for compelling cost benefit analyses.

Allow for tailoring to the types of decisions to be dealt with; don't appear to be calling for a comprehensively detailed analysis in all cases.

Develop a range of scenarios that conceptually represent the test cases for use in task analysis, code analysis, and usability analysis.

Use software tools to prototype rapidly for iterative design. Quality and accountability of software is attractive to program managers.

Focus on prototyping and incremental development.

Pursue tools for component-based programming ("mega-programming").

2.2. FOLLOW-UP INTERVIEWS ON COGNITIVE ANALYSIS TECHNIQUES

25. Since RSG.19 had found little documentation on cognitive task analysis, follow-up interviews were conducted with selected respondents who indicated on their survey they had experience with cognitive analysis. Ten respondents fell into this category. The five most likely to provide the richest information were interviewed.

26. Questions were prepared in advance of the interviews to address
   a) what the domain or task application was,
   b) how information on the task and cognitive requirements was obtained,
   c) whether documentation on the method existed,
   d) what the source of the method was,
   e) how the acquired knowledge was summarised and met,
   f) how successful the method had been.

The interviews lasted from 30 to 60 minutes. Summary and interpretation of the interviews follows.

27. Views of what cognitive analysis consisted of did not always agree. For example, one respondent felt that any consideration of the human dimension implied cognitive analysis. So several of those indicating that they performed a cognitive analysis may have only went to literature on decision making met or tried to establish user requirements. They did not necessarily interact with users or observe their performance.

28. Those developers who had a more accepted view of cognitive analysis offered useful suggestions for knowledge elicitation and engineering. The following are some of the guidelines offered:

   a) Consider using trainers for subject matter experts instead of, or in addition to, operational personnel. Trainers must be well-versed in a domain and must be able to communicate about that domain in order to teach.
   b) Use a preliminary interview to assess the style and level of expertise of the prospective expert. Tailor the knowledge elicitation style to the individual, rather than using a set technique.
   c) Use multiple techniques to investigate different types of knowledge.
   d) Use iterative knowledge elicitation sessions. In follow-up sessions confirm and refine the knowledge organised from the previous session.
   e) Make an audio recording of the elicitation session. This frees the interviewer from taking notes and losing track of the gist of the discussion.
   f) Transcribe the recordings and represent the elicited knowledge. Object oriented software can be used for representation. Consider making two types of representations: one that records the knowledge of a specific expert and another that integrates the representations from the multiple experts interviewed.

29. Some problems with knowledge elicitation and cognitive task analysis were mentioned also in the interviews. When asked how a knowledge engineer might estimate the bounds of knowledge in a domain, one respondent said that any domain is pretty much limitless. What knowledge engineers do is to learn as much as they can within budget and schedule constraints, document it, verify it with the same or different experts, and make interpretations from it. But this can be especially difficult because one of the consistent findings about experts is that their knowledge is so well ingrained or highly abstracted that do not know what they do or are unable to discuss it. Not everyone is a good knowledge elicitor, nor do all domains deal with well-defined subject areas or have clear standards of

performance. In some cases the cognitive task analysis followed the development of the aid rather than leading it. Respondents generally agreed that more efficient methods are needed.

2.3. SUMMARY OF FINDINGS FROM DECISION AIDS SURVEY AND INTERVIEWS

30. Those surveyed and interviewed mostly use decision aid development and evaluation methods that are traditional or fairly ad hoc. Most aid developments are started because of statements of requirement from the customers or sponsors of the aid. Only a few aid developments begin from the developer's own study or recognition of a problem. There was a common feeling that the responsibility for requirements is on the users. This was especially true from respondents with a technical background. (This position seems inappropriate in C2 decision aiding, because of the multi-faceted nature of the task, organisation, and conditions.) Quantitative statements of requirements are rare. The definitions of human limitations being addressed in decision aids are general and indefinite (for example, faster and better decisions).

31. Rapid prototyping is almost always reported as being used, though the prototypes are not always rapid or numerous (for example, 2 prototypes in 6 years). Over a third of the time there are problems with using the feedback from rapid prototyping. Problems include resolving conflicting opinions, comments that are inappropriate for the stage of development, and funding and schedule constraints that preclude any changes from being pursued.

32. About one third of the respondents indicated that they have done cognitive analysis, though follow-up questions and interviews show those methods to be ad hoc and sometimes indistinguishable from more common methods. The methods for analysing decisions were also fairly inexact. Reported methods were either techniques for collecting information or representing it; there were not definite methods for analysing the cognitive aspects of decisions and interpreting them to indicate specific aiding requirements. There were two suggestions about using modelling and bottom-up analysis for decision requirements, but it was not clear what these methods required. Most aids were selected based on user requirements or expert judgement. Sometimes a certain technology or solution approach was the starting point for the aid development, rather than performance or needs analysis. Only in a few cases did there appear to be any deliberate consideration of more than one aiding approach. In these instances, the process of aiding selection and design was described as a creative process of the designer. In another case, experiments were recommended for comparing different candidates for information display.

33. In half of those surveyed, evaluations were informal and often no more than a demonstration with subjective feedback. Multiple evaluations and evaluations by someone other than the developer occurred in over half of the developments. The independent evaluations were mostly not the primary evaluation, but additional user or expert impressions on the aid. Evaluations dealt with purely system issues one fifth of the time. Effectiveness of performance was the most frequent issue on which the aids were evaluated. Remaining issues concerned usability, speed, accuracy, among others.

34. Three fourths of the respondents indicated that existing methods for decision aid development are inadequate. They reported factors that are missing, including accepted theories of C2 and decision aiding, knowledge elicitation methods, cognitive requirements methods, and evaluation and measurement techniques. They felt that development methods should adopt cognitive task analyses. One developer felt that cognitive task analysis should be the starting point and other analyses derived from it. The use of methods should also avoid too much early emphasis on "building decision aids" before front end analyses are addressed. Three fourths of the respondents indicated that they would use methods to better integrate cognitive capabilities, if available. The suggestions for better treatment of cognitive methods have all been addressed in the COADE framework.

CHAPTER 3. SUMMARY OF WORKSHOP ON DRAFT FRAMEWORK

3.1. Goals and Assumptions of COADE
3.2. Cognitive Task Analysis
3.3. Supporting Analyses
3.4. Design
3.5. Evaluation
3.6. General issues

35. RSG.19 held a workshop at George Mason University, Fairfax, VA on June 2 - 3, 1992 to discuss a draft version of the COADE framework. In addition to the RSG.19 members, a group of experts invited from the broad field of decision aid development and three guests participated (see Annex III).

36. The goal of the workshop was to obtain the experts' opinion about the COADE framework: the approach followed and the state of the work at that time, comments and suggestions for additional aspects to deal with, and suggestions for changes or extensions of the framework. To help give a common background for the workshop discussion, experts were requested to provide, in advance, position papers dealing with decision aid development. After the workshop, participants were requested to provide written explanations or examples on selected issues that they referred to during the workshop. Most of the workshop was recorded on video tape and typed transcripts were produced from the discussions. The 200 pages of recorded discussions were reviewed and analysed by different RSG.19 members for their implications for the framework, The analysis was divided into the six topics covered during the workshop:

a) Goals and Assumptions
b) Cognitive Task Analysis
c) Supporting Analysis
d) Design
e) Evaluation
f) General Issues

37. The results of this analysis had substantial impact on the further work of RSG.19. The experts' suggestions were summarised, discussed within RSG.19 and incorporated in the framework with respect to content and format. The most important comments and results from the discussions are summarised below.

3.1. GOALS AND ASSUMPTIONS OF COADE

38. The workshop confirmed the requirement and the potential benefits of a framework for the development and evaluation of complex decision aiding systems. The confirmation was obtained directly, through the experts' answers to particular questions on different issues, as well as indirectly, through analysis of the numerous discussions. COADE could have an important role in establishing methods and techniques for cognitive analysis and design, and in guiding a user in selecting appropriate techniques for all development phases.

39. One of the most consistent comments by the experts was the need for specific examples to elucidate the different steps within the decision aid development cycle described by the framework. Examples would help to show the objective (requirement and purpose) of the activities described and, especially, would improve accessibility, application, and evaluation of the framework. These comments have also motivated RSG.19 to propose the establishment of an Ad Hoc Group (AHG) for evaluation of the COADE approach in new projects (see Annex IV).

40. A further suggestion for making the COADE framework more accessible was to give more guidance to potential users and to point out more techniques. But when asked about the techniques they themselves use, subject matter experts did not give specific examples. All the experts seem to use

their own strategies, individual mixtures of techniques adopted over a long period of time with specific development procedures for specific decision aid developments. In their papers and books they advocate an overall theory for the development cycle, but in reality, each one of the practitioners seems to have his own "cookbook with recipes" for each of the development phases.

41. The experts suggested that COADE should provide an evolutionary guide to decision aid development, useful for developing an understanding about performance problems and potential solutions, but not for specifying a final product.

3.2. COGNITIVE TASK ANALYSIS

42. The experts had different views on whether or not cognitive models are useful. Normative models are not practical, because of the individual nature of each decision situation. Generic models or "concepts" or a common cognitive language would be useful to have in COADE so analysts could describe decision tasks in standard ways. Examples are necessary to show how to apply the concepts.

43. A variety of techniques is available for eliciting knowledge, but there are no standard ways to interpret and apply the information that is acquired. The constraints of the decision situation and the users will greatly limit the set of techniques that are practical to apply. The experts recommended that knowledge engineers be flexible (try a different technique if the current one does not work) and use multiple techniques.

44. The experts felt that performance standards were an important part of identifying deficiencies and targeting design solutions. However developing performance standards is problematic. COADE would be useful if it guided the development of such standards and their use by the analyst.

3.3. SUPPORTING ANALYSES

45. The discussions centred around the notion that cognitive task analysis must be put in context and that supporting analyses (like task analysis, decision makers analysis, and organisational analysis) help provide the context. The larger perspective helps to ensure that the right problem is formulated.

46. The experts said that organisations have a very strong influence on what matters in performance. They recommended that the roles, tasks, information flow, barriers, commander's style, and cultural differences be studied from an organisational perspective.

3.4. DESIGN

47. The design theme of the workshop addressed the issue of how cognitive requirements can be linked, in terms of solution specifications, to decision aid design. The subject matter experts pointed out that there are two types of design. One involves the system architecture and the presentation. The other involves how the problem is fundamentally represented. The cognitive analysis should provide the representation more readily than the presentation. It should be easier to figure out what goes on a screen then how it should look.

48. It was stated that design can proceed by following a list of questions: what is the task, what are the objects, what information should be displayed simultaneously, in what sequence, what cues are important, how do users think about the information? Once these questions have been resolved, then it is a matter of trial and error to derive a design to optimise performance.

49. Another way to overcome typical problems in design is to have better ways of transitioning from requirements analysis to the designer. Designers should be told as much as possible about the cognitive requirements, so they can understand what is meant, not simply what was said. The emphasis in design should be on the motivation of what should be done, rather than how the system

should look.

50. Designers should not have to deal with cognitive requirements separately. Cognitive requirements should be embedded as part of the technical requirements, and the designers should be concerned with satisfying the technical side.

51. There are always going to be design trade-offs, but one of the gains from cognitive task analysis will be to make better trade-offs. It was recommended that the cognitive analysts participate when trade-offs are being made. The success of the design from a cognitive standpoint will be determined by the degree to which the analyst is involved in the whole development process.

3.5. EVALUATION

52. Several issues were addressed in evaluation. Among these were usability evaluations, knowledge base evaluations, software acceptance tests, using established requirements as criteria, "cognitive robustness," structural-functional-purposeful evaluations, and stress testing. A parallel issue was whether the aid improves performance and deciding on the level at which to measure performance (cognitive, task, team, or organisation).

53. There was agreement among participants that evaluation must be an ongoing iterative process, a life cycle issue; that it must be part of the entire development process and that it should provide feedback to the analysis and the design phase.

54. Overall task performance must be the starting point in decision aid evaluation. Overall mission performance dictates the outcomes to be achieved, with processes (strategies, decision etc.). Factors to be taken into account in conducting an evaluation include the sample, the aid itself, the scenarios, and the measures. Also, the point at which an evaluation is conducted within the life cycle dictates what should be measured.

55. Decomposing the evaluation itself into phases might be a fruitful approach, permitting the particular measures of interest that must to come out of the analyses to be specified at each phase. Among the different methods or techniques proposed for evaluating decision aids were storyboarding (a type of walk-through technique), computer-based programs and scenario-based techniques. A "waterfall" approach will assess each step of the development cycle by evaluating whether it conveys the former step accurately; a "failure-driven" approach evaluates the decision aid by designing scenarios in which it might fail.

56. Several experts agreed that the scenarios selected for evaluation are crucial to the outcome. In particular, testing an aid over a range of scenarios can indicate its "robustness". Scenarios that are likely to make the aid fail may help to evaluate its effectiveness, but can be dangerous if they are unlikely to occur.

3.6. GENERAL ISSUES

57. General issues addressed the format and implementation of the COADE framework. Several suggestions for improving the comprehension and usability of the framework emerged from the discussions. One purpose of COADE is to get users to see their role in analysis-design-evaluation differently, to have them view their decision making behaviour more deliberately, and to give them leverage as part of a design team.

58. Experts recommended changing the COADE description so that it will come "alive". This can be done by giving the users an explicit vision of what they will be able to accomplish: by including examples of the products, including a knowledge elicitation dialogue to show a method, and using stories, pictures and tables to simplify information.

This page has been left blank intentionally

CHAPTER 4. INTRODUCTION TO THE COADE FRAMEWORK

4.1. Introduction
4.2. Overview COADE Activities
4.3. ANALYSE
4.4. DESIGN
4.5. EVALUATE

4.1. INTRODUCTION

59. The goal of COADE is to assist in the development of decision support in complex systems, based upon knowledge of the human's role, capabilities, and the tasks to be performed. COADE provides a framework for

a) the identification of cognitive requirements of the tasks;
b) the development of design requirements that address those cognitive requirements;
c) the assessment of the quality of intermediate and final products or results in the developmental process.

60. These three components of decision support development are addressed via three sets of activities: ANALYSE, DESIGN, EVALUATE:

a) In ANALYSE, the goal is to identify the critical aspects of the system. Analysis results in a specification of critical cognitive requirements on which design decisions should be based.
b) In DESIGN, the goal is to find potential solutions for the identified problem and to develop ways of interacting with computer-based solutions.
c) In EVALUATE, the developer steps aside for a moment and verifies the results of the analysis and design activities in a more or less formal way.

4.1.1. The Human in the System

61. The perspective one takes on system development determines the nature of the resulting system requirements. COADE takes a human-centred systems perspective. A system is defined here as comprising humans and machines working together to achieve operational objectives. The human is not merely a component of the system, but has control over the actions and has the responsibility to attain the organisational goals. A human-centred approach deviates from engineering approaches that see the alleviation of technology deficiencies as the core goal of system development. In an engineering approach the human is seen as being necessary in the system to fill in the holes that could not be filled with technology. In a human-centred approach to system development, the central premise is that systems exist for human purposes (Gardner, 1991; Rouse, 1991). This implies that:

a) the objective of system development is to support humans in achieving the operational objectives for which they are responsible;
b) human characteristics should be used as a basis for the analysis of current and future systems;
c) criteria relevant to human performance should be included in the evaluation of the effectiveness of a system.

62. A cognitively-centred approach builds upon a human-centred approach and zeroes in on the cognitive requirements of the system and their consequences for system design. The focus of COADE is on the cognitive processes in complex systems, such as C2 systems. It supports the understanding of the mental processes, goals, and knowledge involved in C2 tasks, the identification of limitations in the cognitive performance on the tasks, the identification of opportunities for support of the decision making processes, and the development of multiple potential solutions and interface prototypes.

4.1.2: System Development Steps

63. System life cycle methodologies support the management of the development of complex systems that involve hardware and/or software components (Department of Defense, 1988). Several methodologies for developing systems have been proposed (Sage, 1992)[1]. Most methodologies give little attention to a human-centred and cognitively-centred approach. Andriole (1990) has proposed a system development structure that incorporates behavioural and cognitive input into system development. Beevis, Essens, and Mack (1994) discuss a human factors engineering plan for CCIS projects that provides a context for cognitive analysis activities. Requirements definition, including human and cognitive requirements, should be defined before system design starts (Andriole, 1990). DOD-STD-2167A identifies eight phases in a systems software development, together with deliverables, evaluation criteria, progress reviews and criteria for evaluating deliverables. COADE activities match this process: ANALYSE starts in the early phases (as early as concept development); DESIGN runs from conceptual design on to detailed design; EVALUATE parallels the whole developmental process (Figure 4.1). ANALYSE continues after requirements specification, if new aspects of the system are discovered as a result of evaluation of concrete design concepts and prototypes.

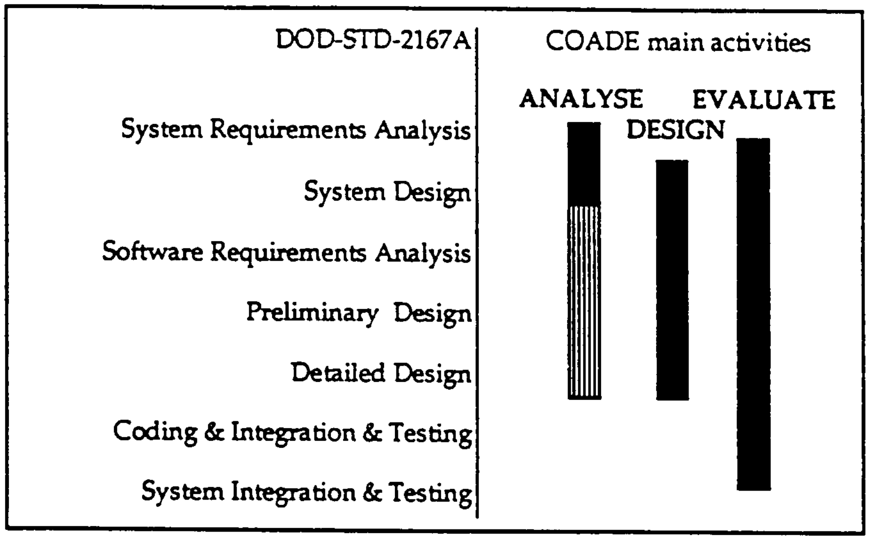

Figure 4.1. Relationship between COADE Activities and DOD-STD-2167A System Development Phases

64. NATO Defence Research Group Panel 8/RSG.14 has suggested a sequence of development steps parallel to the 'standard' system development steps that will help to define the human role in a system (Beevis, 1992). Central to their approach is the analysis of functions and tasks to be performed in the system by either humans or machines. They recognised, however, that the majority of function and task analysis techniques do not lend themselves to the description of cognitive behaviour. The description of behaviour is done mainly in terms of the observable activities. COADE addresses the description of mental activities, and focuses on the knowledge required and used in performing tasks and the mental processes that operate on this knowledge. However, cognitive processes do not occur in isolation, but result (eventually) in observable actions. COADE will refer to the techniques described by RSG.14, in particular task analysis techniques, as means of specifying the structure and timing of the overt behaviour in the execution of tasks.

65. The RSG.14 model leads from mission analysis and function analysis to task analysis. A function is a logical unit of the behaviour of the system, but one that is not yet allocated to a particular element of the system (machine or human). Tasks are logical units of the human behaviour in the

[1] For an overview, see Sage, 1992. This book on Systems Engineering is a good source for issues in the design of large-scale systems intended for decision support.

system. Function allocation is relevant for the design of systems in which there are gross functions that can be assigned to machines or computers. The criteria for assigning these functions to the human are based upon what humans can do well and what their limitations are. The functions assigned to humans are candidates for further analysis concerning support. COADE is well-suited to identifying support needs. Eventually, information about the human cognitive functioning may be turned into function allocation criteria.

4.1.3. Iterative Development

66. Experience tells us that the development of systems is not a straight forward process. System development can be characterised as the solving of an ill-defined problem (Cody, Rouse & Boff, 1993; Rouse & Boff, 1987b). It is problem solving, because it is not immediately clear how to get to the solution (the system). The problem is ill-defined, because it is not clear what the goals are and what constitutes a solution. This differs from well-defined problems, e.g. a puzzle, in which the end state (goal) is defined and each step in the development towards the solution can be compared against the goal (means-end analysis). With an ill-defined problem any of several solutions could constitute a possible end state. Direct involvement of the decision makers in a dynamic simulation of tasks and situations ('man in the loop') is necessary in order to come to a conclusion as to whether a solution adequately supports the decision making processes.

67. An expensive, but direct approach is to build the system completely, evaluate it and, if it is not correct, rebuild it. A more efficient approach is the prototyping approach that allows iterative and incremental growth of the solution (Boar, 1984). A prototype is a tentative, intermediate, and tangible representation of a (partial) solution to a problem built for the purpose of testing the validity of the assumptions on which the representation is based upon. A prototype gives the developer the opportunity to show his or her understanding of the problem and gives the stakeholders the opportunity to steer or correct it. Andriole (1990) claims that it is impossible to validate analytically-derived user requirements without a working model of prospective system capabilities. Prototyping represents a strategy that uncovers design flaws, costly issues and invalid requirements <u>before</u> expensive programming begins. A prototype provides the stakeholders with an understandable view of the system's functions. It gives the stakeholders a say over the product and makes them accountable for the achievement. It reduces the risk of misunderstanding between user and developer.

68. There are, however, fundamental problems with the prototyping approach. Several are mentioned here. There is a risk that, intentionally or not, prototyping permits too easy a jump to a 'favoured' solution, a solution that is not necessarily wrong, but may not address the critical issues in the system. The prototype, because of its visibility, may allow the developer to more easily bypass consideration and comparison of several solutions, and fixate on it as the "easiest" solution, especially when time is short (Klein, 1987). Prototyping can easily lead to a technology-driven definition of the solution, whereas the development should be requirements-driven. A significant and systematic user involvement is required in proper prototyping, with users that are willing to invest in changing their way of work. It is not enough to evaluate the prototype with the expert users who were involved in the definition of the prototype; 'fresh' users must be used for judging the functionality and usability of the system as it evolves. The evaluation of prototypes requires a systematic approach which entails more than just showing it to users to elicit impressions and casual comments. The effort needed to repeatedly adjust the prototype can provoke resistance in the developer.

69. ANALYSE and DESIGN are connected in an iterative cycle, in particular, when the allocation of functions to a machine or computer (a result of DESIGN considerations) has a major impact on the organisation of tasks. The new allocation of tasks must be assessed to see what consequences it has for performance.

70. EVALUATE activities performed during analysis and design also provide a basis for iterative development. Feedback is sought from the stakeholders of the system during evaluation, using modelling, simulation, and prototyping techniques. COADE advocates that prototyping should be requirement-driven. This prevents solutions that come out of the blue. Computer-based prototypes that are intended to reflect the future computer functions are seen as particularly relevant for the design phase in which requirements are transformed into solutions and more detailed requirements concerning the human-computer interface are specified.

4.2. OVERVIEW OF COADE ACTIVITIES

71. Figure 4.2 gives an overview of COADE activities and products, together with the reference information concerning relevant concepts and models.

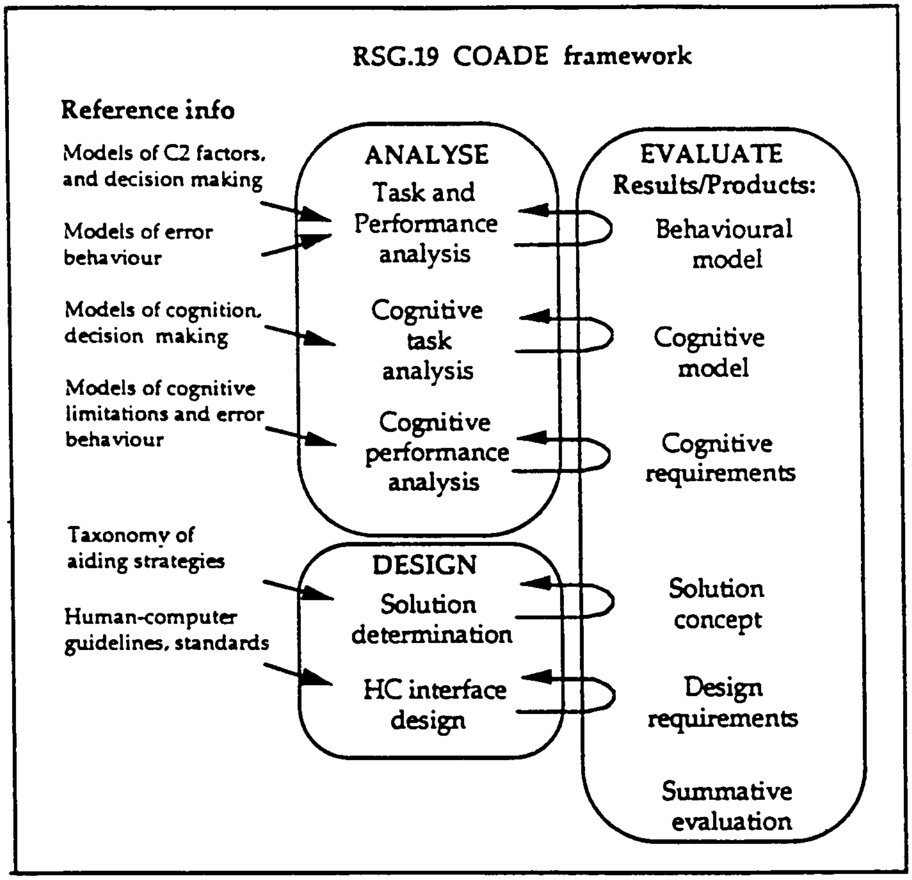

Figure 4.2. COADE Framework with Reference Information, ANALYSE, DESIGN, EVALUATE Activities and Their Products

4.3. ANALYSE

72. ANALYSE aims at generating cognitive requirements based on a detailed analysis of the critical cognitive tasks required to attain the operational goals of the system. The startpoint of the analysis can be the set of tasks which resulted from the mission and scenario analysis and initial function allocation (Beevis, 1992). Two representations of the human activities in the system are generated: a Behavioural Model, which represents the goal structure of the task, and characteristics of task, person, and environment; and a Cognitive Model, which represents the knowledge and mental processes applied in the tasks. The specification of the cognitive requirements that need to be addressed in system design completes the ANALYSE activity.

73. The relationship between the main activities in ANALYSE and their products is shown in Table 4.1.

Table 4.1. Relationship between COADE ANALYSE activities.

	Description of the Task	Assessment of Performance
Behavioural factors (results of performance)	Task Analysis: What are the significant behavioural requirements of the task? What are the inputs and outputs?	Performance Analysis: In what ways is performance limited? Which aspects of performance are most deficient or opportune?
Cognitive factors (knowledge and skills)	Cognitive Task Analysis: What are the significant cognitive components of the task and the behaviours? What are the processes, the transformations?	Cognitive Performance Analysis: In what ways is cognitive performance limited? Which aspects are most frequently limited, most critical? Is it feasible to remedy the cognitive limitation?

4.3.1. Task and Performance Analysis

74. Among others, the work of NATO DRG Research Study Group 14 on 'Analysis techniques for man-machine system design' gives the background for techniques, such as mission and scenario analyses, that provide information about the overall requirements of the system under development (Beevis, 1992; see also Kirwan & Ainsworth, 1992). The development and analysis of critical mission scenarios, in particular, is a very effective way of (normatively) describing functions and tasks that are performed in operations. Scenarios can serve as the backbone of discussions with the stakeholders and users about the system under development.

75. Task analysis addresses three components of the task: the task characteristics, the person characteristics, and the environment characteristics. Task analysis gives the context of the events and required actions in the system against which the cognitive analysis can be positioned. The decomposition of the goal structure of the task in terms of overt activities provides the focal points for the cognitive processes, and it anchors the cognitive activities to the events in the system. Task analysis identifies the operational performance criteria, how the task should be performed on several dimensions such as time constraints, frequency, and accuracy. Time line analysis shows the temporal organisation of the task e.g. represented in the form of an Operational Sequence Diagram.

76. Analysis of the person characteristics (decision maker analysis) provides the level of expertise, general intelligence, styles and attitudes of the expected users. Environmental characteristics refer to the conditions under which tasks have to be performed, stress, noise, vigilance, individual vs. group, team organisation.

77. The analyses result in a description of what should be done together with factors that may impact the performance of the system. The question of how well this is done is addressed in the Performance analysis. Performance analysis identifies how well tasks are required to be performed and how well they are actually performed. The performance deficiencies so identified require further analysis to determine the sources of deficiency. The deficiencies should trigger modifications to the system to improve the match between the actual and the required performance.

4.3.2. Cognitive Task and Cognitive Performance Analysis

78. A cognitive task analysis produces a description of a task in cognitive terms. It describes how memory is involved for temporary storage of information; what knowledge, facts and procedures

are involved; what strategies are used to solve task problems; what deductive and inductive transformations are applied to available information; what goals people formulate explicitly and implicitly to achieve their objectives; what judgements have to be made; and what decisions have to taken. Through interaction with problems in a task humans learn how to deal with them, that is they acquire expertise. Mental structures that guide in task problem solving and inferencing are developed. Cognitive task analysis taps those structures to obtain a cognitive model of the tasks carried out by humans. Although observation is essential for obtaining a feel for the context in which cognitive tasks are performed, special techniques are required to assess the expertise on which cognitive task performance is based.

79. Knowledge elicitation (KE) techniques are used to tap cognitive processes and elicit the knowledge structures and content used in the task. Knowledge comprises both factual knowledge and the procedures for applying that knowledge in solving and executing task problems. Some KE techniques stress the concepts used, others focus on the strategies that control the cognitive behaviour. A mix of techniques is advised.

80. Cognitive Task Analysis results in a description of <u>what</u> knowledge is involved and of the processes that are assumed to operate. The question of <u>how well</u> people perform cognitively is assessed in Cognitive Performance Analysis. The purpose is to judge the acceptability of the cognitive performance in order to determine the aspects of performance that are critical or that could be improved. The analysis results in the formulation of cognitive requirements. Cognitive requirements are those aspects of human cognitive performance that are critical for the functioning of the system. They must be addressed in system development and form the criteria against which future developments can be assessed.

4.4. DESIGN

81. The purpose of DESIGN is to translate cognitive requirements into system design specifications. In the previous set of activities the question was <u>what</u> should be supported or accommodated. During the initial analysis, the answer to that question should not be constrained by what the developer thinks is possible. At this stage, however, the focus is on <u>how</u> to provide the required support. Now the aim is how to realise the requirements, to integrate them, and to determine a balance between requirements and costs.

4.4.1. Solution Approach Determination

82. Often a solution concept is established early on in the project, e.g., a computer-based solution is expected to solve all problems. Often these early choices are not based on solid analysis but on personal preferences. The purpose of the solution approach activity is to establish the best approach for correcting identified problems, e.g., different procedures, training, or personnel. A second purpose is to identify the impact the solution has on the overall system in terms of additional costs and other requirements (e.g. maintenance).

4.4.2. Aiding Concept and Design Requirements Configuration

83. There is no one-to-one mapping between cognitive requirements and solutions. The transition from cognitive requirement to design solution is often a creative process, which requires an iterative development of ideas. Multiple ideas may be pursued in parallel. COADE provides a list of aiding strategies and guidelines to support the solution generation process.

84. The iterative development of ideas requires a systematic evaluation of the proposals with users to verify the adequacy of the solution. Prototyping tools support the fast turn around of ideas and user feedback.

4.4.3. Human-Computer Interface

85. When a computer-based solution is being considered a detailed specification of the interaction between human and computer must be developed. The design of the interface between user and computer is a complex and subtle matter. Emerging standards and user interface management systems define certain aspects of the interface, but the critical issue is to understand the goal structure of the user at this level of execution of the task and to provide adequate feedback in the dialogue.

86. The development of the human computer interface follows a cycle of analysis, design, and evaluation. During analysis, the (higher level) descriptions from the cognitive task analysis are detailed further to capture the goal structure of the user. The question at that point is what to present. During design, the question is how to present it. Representations of the information on the screen and feedback concepts are created and tested with the user. The interactive nature of the dialogue requires extensive user involvement in verifying the adequacy of the dialogue.

4.5. EVALUATE

87. Evaluation of systems is considered to be a complex matter and is often overlooked in actual development. There is a tendency to cut down on evaluation effort, because it is usually scheduled at the end of a development ('build and test') when time and money run out. The goal of evaluation is to ensure the quality of what is produced and to prevent unpleasant surprises from arising at the end of the project when they cannot be corrected. To ensure evaluation as a crucial element of system development, COADE integrates it with the analysis and design activities. The evaluation activities provide an ongoing verification and validation of products during system development. Products that must be evaluated are task models, task specific cognitive models, conclusions concerning critical factors, cognitive requirements, solutions, interfaces, and prototypes.

88. The quality of the ongoing process must be assessed with the stakeholders (including direct users) of the system. The purpose is to get their agreement concerning the developer's understanding of the problem. The forum of stakeholders should have the authority to give directions and feedback to the developer and the analyst. This forum provides one opportunity for evaluation, particularly at the early stage of the development, but must not be a substitute for evaluation with the actual user group to assess the quality of the cognitive models and the more detailed cognitive requirements. As the design concepts become more tangible users can 'work' with the design and provide feedback from their dynamic interaction.

89. A distinction is made between interactions with (expert) users during analysis and design and during evaluation. During analysis and design the intention is to develop and generate concepts and models and identify critical issues. Interaction with the user during evaluation has as its purpose the more formal review and judgement of the product. COADE supports the systematic involvement of the user.

4.5.1. Evaluation during ANALYSE

90. Verification of the models developed includes assessment of the accuracy and completeness of the models, requirements, and performance measures. Criteria for these assessments should be developed during ANALYSE.

4.5.2. Evaluation during DESIGN

91. Verification of the solution proposals concerns the question of whether the proposed design actually addresses the requirements specified in the ANALYSE activity. Prototyping is seen as the primary vehicle for evaluation. However, it is not sufficient to confront a user with a prototype and ask for comments. A systematic approach is necessary, using a task scenario run with the prototype.

4.5.3. Summative Evaluation in EVALUATE

92. There must be a final statement concerning whether the system actually accomplishes its goal. Also, the integration of a new or modified system with other systems should be assessed. User acceptance plays an important role in the success or failure of a system. To assess this comprehensively, the system should be evaluated in a fielded mode and evaluations continued intermittently throughout its application life.

CHAPTER 5. COADE FRAMEWORK

5.1. ANALYSE
- 5.1.1. Task Analysis
- 5.1.2. Performance Analysis
- 5.1.3. Cognitive Task Analysis
- 5.1.4. Cognitive Performance Analysis

5.2. DESIGN
- 5.2.1. Solution Approach
- 5.2.2. Human-Computer Interface Design Activity

5.3. EVALUATE
- 5.3.1. Evaluation
- 5.3.2. Evaluation Activities during ANALYSE
- 5.3.3. Evaluation Activities during DESIGN
- 5.3.4. Summative Evaluation Activities

This page has been left blank intentionally

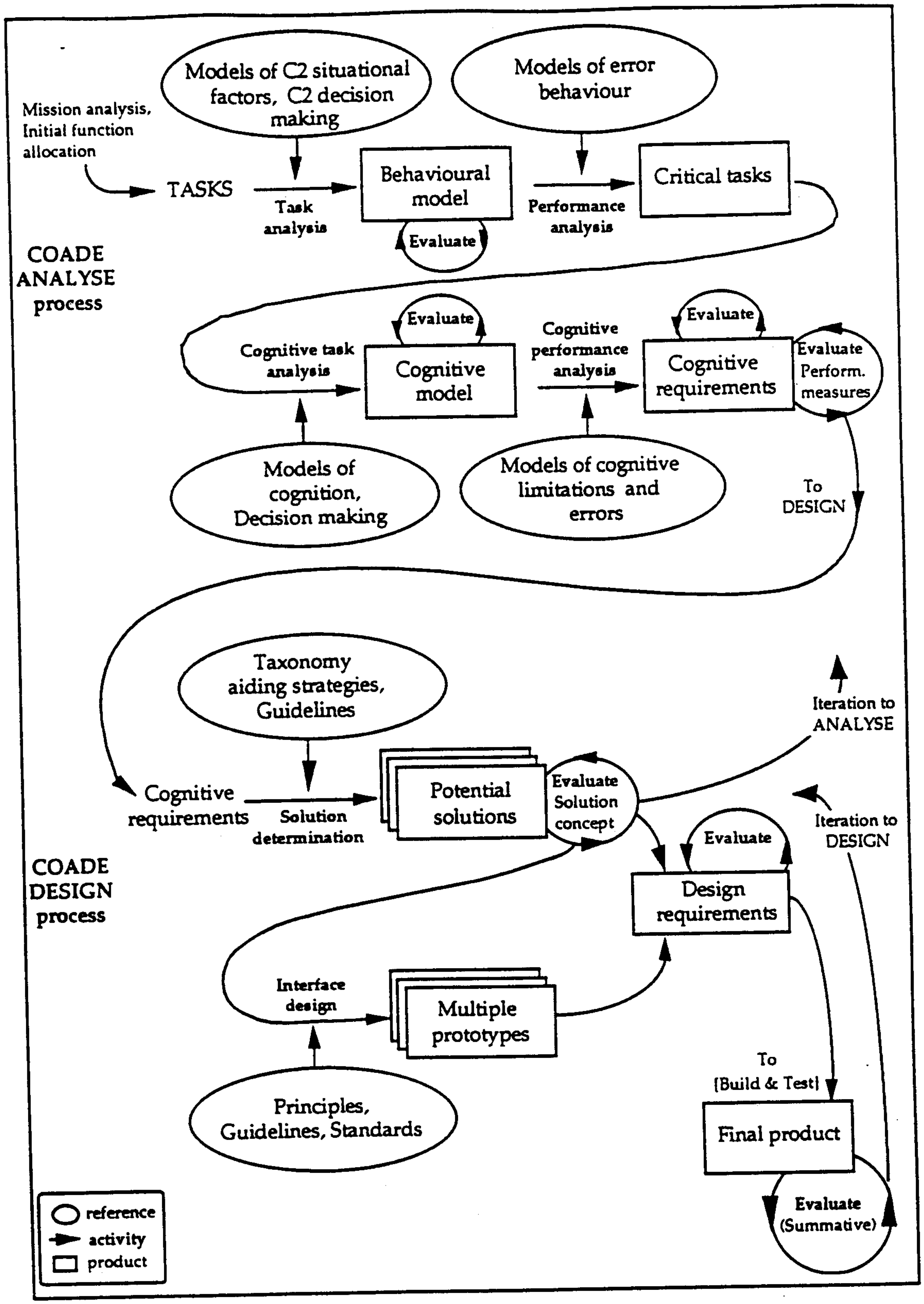

Figure 5.1. COADE Model of Cognitively-centred System Design: Relationships between Activities, Products and Reference Information,

5.1. ANALYSE

5.1.1. Task Analysis
- 5.1.1.1. Decision Makers and Organisational Analysis

5.1.2. Performance Analysis
- 5.1.2.1. Performance Measures (R.1)
- 5.1.2.2. Performance Standards (R.2)
- 5.1.2.3. Types of behavioural errors (R.3)

5.1.3. Cognitive task analysis
- 5.1.3.1. Knowledge Elicitation (R.4)

5.1.4. Cognitive Performance Analysis
- 5.1.4.1. Cognitive limitations (R.5)

5.1.1. Task Analysis

Description.

93. The task analysis produces information about the task and related aspects. Related analyses deal with decision makers, the organisation, and performance. The COADE framework describes the activities for task analysis, decision makers analysis, organisational analysis, and performance analysis. Task analysis also includes information on the situational context (mission analysis), functional analysis, and existing system components. Functional analysis is a technique which allocates functions among system components (like, human operators, computer subsystems, mechanical subsystems, communications, etc.). The task and supporting analyses provide the necessary goals, constraints, and conditions in which to consider the cognitive aspects of performance. Task analysis produces a detailed description of the way a task is or could be performed. It produces a detailed description of

a) Why the task is performed (goals).
b) How it should be performed (procedures).
c) What external resources are required for performance (such as, information, subsystems).
d) Under what conditions (normal, unusual, extreme, or dangerous) the task is performed and external events affecting the task.

94. The information from the task analysis will provide necessary background information for identifying task performance and generating a model of behavioural performance.

Products.

95. The task analysis in COADE provides descriptive information to construct a behavioural model of the task. Task analysis identifies the objects and structure for the behavioural model. It offers insight into important issues to focus on in behavioural modelling. Performance analysis adds indications of how well tasks are performed.

Methods.

96. There are many approaches to task analysis. Sources describing task analysis include Beevis (1992); DeGreene (1970); Drury, Paramore, Van Cott, Grey and Corlett (1987); Laughery and Laughery (1987); and Meister (1985).

97. Meister indicated that task analysis is useful for determining changes in tasks that will require changes in personnel, selection, training, and system operation and for determining whether personnel will be able to perform all functions effectively. He identified compatible behavioural methods to assist in addressing system development questions. The methods included analyses for missions, functions, function allocation, workload (see Lysaght, Hill, Dick, Plamondon, Linton, Wierville, et al., 1989), information, decisions, time lines, linkages among components, and errors and failures (see Jones & Grober, 1962). COADE clusters these together as task and performance analyses and adds analyses on decision makers and organisations.

98. US MIL-STD-1478, Task Performance Analysis (Department of Defense, 1991; Miles & Geddie, 1989), identifies information to be obtained for a task analysis. A list of information requirements for task analysis adapted from MIL-STD-1478 by Fallesen and Quinkert, 1990 follows:

a) Task goals
b) Task procedures
c) Information required, including information flow and the form of the information
d) Actions, operations, decisions
e) Knowledge and skill requirements
f) Task and performance dimensions
   criticality; frequency; accuracy; work rate; time constraints
g) Intervening or moderator factors on performance
h) Simultaneous or prerequisite tasks
i) Product produced or result.

99. US MIL-STD-1478 also has a list of action verbs for describing or categorising tasks.

Fleishman and Quaintance (1984) provide lists of human abilities. For example:

Verbal comprehension	Verbal expression
Ideational flexibility	Originality
Memorisation	Problem sensitivity
Mathematical reasoning	Number facility
Deductive reasoning	Inductive reasoning
Information ordering	Category flexibility
Spatial orientation	Visualisation
Speed of closure	Flexibility of closure
Selective attention	Time sharing
Perceptual speed	(plus 18 motor abilities)

100. Descriptions of missions should be available by the time task analysis starts. If not, missions from previous systems or similar tasks can be adopted. The mission information provides the conditions for tasks and bounds the problem. Narrative descriptions and graphic mission profiles are two formats of mission information (Beevis, 1992). Narrative mission descriptions consist of detailed descriptions of mission events (e.g., sequences, times, mission constraints, and environmental conditions). Graphic mission profiles show a sequence of operational events or situations, which help to determine the functional and performance requirements of the system. Factors in a graphic profile might include system states, geographic locations, and movement rates. Mission analysis can be used to project the most likely or most dangerous contingency situations. The following mission or situational features are suggested to be identified for the task analysis:

a) Level of decision making (strategic-NATO, allied, national)
b) Tactical echelon and size of force involved
c) Geographic area, terrain, climate
d) Nature of threat
e) Role of the force (heavy conflict, medium intensity, low intensity, peace-keeping, nation building, counter-terrorism)
f) Expected durations of missions
g) Phase of operation (peacetime, pre-hostilities, deployment, execution, sustainment, consolidation, transition to peacetime, withdrawal).

101. An allocation of functions may also be available when beginning a task analysis. Functional allocation assigns functions to humans and subsystem components. Rouse (1991) lists three approaches: comparison (the superior performer is assigned the function), leftover (as many functions are automated as technology permits), and economic (allocations are based on the costs involved in automating functions). RSG.14 (Beevis, 1992) also describes methods for functional analysis and allocation. Once the design of an aid begins, functional allocation becomes an iterative process, assigning functions and tasks depending on the identification of capabilities and limitations of representative decision makers and decision aiding technology.

102. The basic techniques involved in compiling the desired task information include using documentation or knowledge from similar or predecessor systems. Various ways of representing task information include operational sequence diagrams, information flow diagrams, task hierarchies, decision trees, task network dependencies, concept maps, and narrative lists. Selection of a specific representation technique will depend on the amount of detail desired, the complexity of the task, and whether a computer version of the behavioural model will be developed. The Taxonomic Workstation (TWS) (1990) is a software tool for the creation and manipulation of taxonomies. It has a relational data base with pre-stored lists of functions, tasks, subtasks, abilities, and skills. It can be used to build task and ability inventories for specific applications.

103. Characteristics of decision makers and the organisations in which they work are important for assessing task procedures and information needs. Social or individual traits can differ among people. The traits may provide explanations for differences in observed or effective performance, e.g. why some individuals are more successful at some tasks than others. Organisational factors influence individual goals and values and are especially important for establishing standard work practices. Separate activities are described for decision makers and organisations (see Section 5.1.1.1)

104. Since task analysis requires considerable effort, the analyst must carefully choose those areas most critical for representing and distinguishing among performance levels. Carter, Archer, and

Murray (1988) describe a process of prioritising command and control tasks as candidates for planning aids. Their approach includes experts' ratings of task importance, complexity, required level of effort, criticality, time constraints, required special expertise, quantity and availability of data, and any other special factors.

Relationship to other activities.

105. Task analysis is a very broad, yet fundamental part of the design of any system involving humans. Task analysis provides background information to construct behavioural models and these models are used to identify performance limitations and capabilities. Task analysis (including information on missions, functions, information, decision makers, organisations, and performance) also provides conditions and constraints important for the design, evaluation, and implementation of decision aids. Task analysis should also be used to determine how a task will change after an aiding component has been incorporated (Garg-Janardan, Eberts, Zimolong, Nof & Salvendy, 1987). This is addressed in Section 5.2.1 'Solution approach' of the design phase.

Rationale.

106. Task analysis is an integral part of system design. It is likely that task analysis information may already be available from a related system or the overall system in which a new aid will be incorporated. If so it can be applied to the construction of a model of task behaviour. Consideration of task performance is necessary to analyse cognitive performance. Thorough understanding of task analysis methods and knowledge of example applications should be a basis for the analyst performing a task analysis.

5.1.1.1. Decision Makers and Organisational Analysis

Description.

107. The decision makers and organisational analysis identifies important characteristics of the personnel who perform required tasks and the nature of the organisation in which they work. Information on characteristics of decision makers and organisations provide performance effects for the behavioural task models and the subsequent use of the models.

Products.

108. The decision maker-organisational analysis should produce information that describes human- and organisation-unique effects on performance. Categories of information that are of interest on decision makers include:

a) Number performing the tasks;
b) Jobs, positions, duties, or roles;
c) Typical knowledge, skills, and abilities;
d) Levels of experience;
e) Attitudes, values, and motivations;
f) Thinking styles;
g) Range of performance.

109. Categories of information that are of interest on organisations include:

a) Structure and size of groups;
b) Distribution of authority and responsibility;
c) Socialisation and cohesion;
d) Attitudes, values, and motivations;
e) Organisational performance norms (expectations);
f) Adaptability to changing situations, different problems;
g) Performance capacities.

Methods.

110. Descriptive information should be assembled about the effects of individual and organisational characteristics on system performance. These conditional parameters can be used to more fully describe behaviour for the task model. With an adequate description of the influences of individual and organisational differences a better model of performance can be established.

111. Some of the categories provide physical information for the structural aspects of the model (e.g., the number of decision makers and support personnel, the kinds of hierarchical structures). This type of information can come from surveys of specific systems or standard organisations functional and structure documents. A target audience description (TAD) is a formal statement of the required skill levels, numbers of personnel, anthropometric ranges, and training needs of personnel (Booher, 1990). While techniques like the early comparability analysis (ECA) (1987) exist and quantitative and qualitative personnel requirements inventory (QQPRI) (Bryan & Regan, 1972) exist, they are not especially suited for jobs requiring complex decision making. Documentation on predecessor or reference systems can be used for estimates on the number and types of personnel needed.

112. The more qualitative factors (like attitudes, cohesion, and adaptability) should be identified and considered in terms of how they influence performance processes and outcomes. See Section 6.4 for a more detailed discussion of these factors. Interaction with experts and literature searches are the primary means to determine the effects of various factors on performance.

Relationship to other activities.

113. These analyses are part of task analysis and help to describe the predisposition of behaviours (i.e., decision maker characteristics) and the contextual demands for the situation. Decision maker characteristics can be useful for

a) establishing performance standards,
b) selecting or judging the quality of representative decision makers for knowledge elicitation,
c) basis for decision aiding (e.g., expert knowledge, decision strategies),
d) accommodating different styles of information processing or decisions making.

114. Organisational characteristics provide the broader context in which to examine performance for models, analysis, and evaluation. Constraints placed by the organisation may help identify what is possible in terms of decision aiding or may introduce special requirements that the aid has to meet.

Rationale.

115. Without an analysis of decision makers and organisations there will not be the proper delineation of the range and predominance of behaviours, experiences, attitudes, constraints, etc. For example, the analysis might assume that the decision makers use an analytical style of problem solving and so decision theory is selected for aiding the task. Instead the decision makers in a certain situation may use a global, intuitive approach. A behavioural model assuming deliberate, analytical procedures would not produce very accurate analysis of performance.

116. The Framework focuses on C2 decisions from the perspective of the individual, but C2 occurs in the context of organisations. Because the focus of the Framework is at a cognitive level, it is appropriate to emphasise individual decision making. But the Framework also recognises the need to consider the various influences on the individual. Organisations influence decision makers in a number of ways. Framing the behavioural problem from an organisational perspective will develop a better understanding of the goals, resources, and constraints which contribute to decision making.

5.1.2. Performance Analysis

Description.

117. The purpose of performance analysis is to complete the behavioural model of the task and to use the model to identify task deficiencies. Insight is achieved on how well tasks must be performed and what actual or expected deficiencies there are in unaided performance. The performance analysis activity assesses performance to identify problems or to project performance capabilities. Judgement of performance is based on comparison to a standard (existing or specially-derived, explicit or implied).

Products.

118. The results of performance analysis are a behavioural task model and the identification of behavioural performance capabilities and limitations for the task.

Methods.

119. The behavioural model can consist of descriptive information about the structure of tasks, people doing the tasks, human abilities and processes, functional allocation, organisational structures of the people involved in the task, information flows, and so forth.

120. A model generally represents two types of entities. There are elements that represent concrete or tangible items, events, and concepts. These include things like organisational units, number of personnel, amount of time available to do a task, information reports, and equipment functions. Also more abstract concepts, relationships, and principles can be represented. These subjective issues include things like the criticality of information, levels of expertise, styles of problem solving, degree of risk-taking, amount and effects of stress, adaptability of the group to change, and so on. These "subjective" issues are the most challenging to represent in behavioural models, because of the difficulty and uncertainty determining what they should be and how to represent them. (For example, should the model represent only high stress conditions or should stress be free to vary randomly, vary by some other distribution, or vary based on conditional events.)

121. Ramsey and Atwood (1979) identified four basic modelling approaches related to humans and tasks:

a) Network models represent series of tasks in a logical predecessor-successor relationship. Schweickert and Fisher (1987) provided an updated review of network models using mathematical techniques to represent mental processes.
b) Control-theory models typically represent tasks as a control sequence. Wohl, Entin, Kleinman, and Pattipati (1984) discuss various control theory models for command and control tasks.
c) Decision theory models represent various "states of the world" and the values, probabilities, risks, and costs associated with various courses of action.
d) Human information processing models represent the task environment, the problem solver's representation of the problem, and the procedure to arrive at a solution. Card, Moran, and Newell (1986) developed the Model Human Processor that principally uses a recognise-act operation between working and long-term memory.

Meister (1985) and Rasmussen (1986) both provide further descriptions of modelling approaches. Refer to Crumley and Sherman (1990) for a review of some 40 specific examples of descriptive C2 models (like organisational processes, behavioural decision making, information processing, and network models).

122. Simulation tools, that are usually supported by computers, can be used for providing a working representation of a behavioural model. SAINT (system analysis of integrated network of tasks) (Meister, 1985), Micro Saint, Petri nets, Human Operator System (HOS) V (Plott, Dahl, Laughery & Dick, 1989), and Task Analysis Workload (TAWL) (Bierbaum, Fulford & Hamilton, 1990) are examples of computer modelling tools. A task model can also be represented using paper simulation, a narrative description, or other non-computer forms. A set of small paper or computer models can be used to explore a performance issue in more depth or simply used as an alternative to a complex, larger model.

123. The behavioural task model serves as a tool to analyse performance. It does not produce

analysis results without human intervention. The model can be used to represent behaviours, gain insight into relationships, and assess specific performance problems. The model's ability to provide insight into problems depends on three basic elements: measures, standards, and data. Successful analysis must be based on good performance measures, adequate standards, and reliable and valid indicators of actual (or projected) performance. Data can be collected to describe performance, determine expected levels of performance, or identify associated limitations in performance. The following Sections (5.1.2.1 and 5.1.2.2) discuss types and qualities of performance measures and the derivation of performance standards.

124. Knowledge about previous performance and the anticipation of changes can be used to develop a performance standard. Actual performance data or projected performance levels are compared to the standard to identify degrees of discrepancy. Data can be compared to the derived standards, but the analyst should also be looking for extremely poor performance or erroneous performance which needs to be avoided. Error analysis (Rasmussen, 1982; Reason, 1990; Rouse, 1990) is used for the prediction of errors. The common element of these approaches is to predict what form errors will take and under what conditions they can be expected. Insight into the conditions is achieved with descriptive information from the task analysis and the behavioural model. Predictions of errors are based on the comparison of standards to observed performance or to projected performance results from the model. Examples of error forms are given in Section 5.1.2.3.

Relationship to other activities.

125. Task analysis provides information on task goals, required information, procedures, and other conditions of task performance (organisation, personnel, environment, mission, threat, equipment, etc.). Reference information is provided on measures and standards. Data are collected and used in the performance analysis.

Rationale.

126. Distinctions can be made among performance analysis, traditional task analysis, and cognitive task analysis. Task analysis is often more descriptive of "how to" perform a task rather than "how well" (or poorly) a person performs. Cognitive analysis goes into greater depth on the cognitive processes. Performance analysis should produce an accurate reflection of how well tasks are performed or, from a projective viewpoint, how well they might be performed. Performance analysis is important because task analysis sources typically do not indicate how to assess performance.

5.1.2.1. Performance Measures (R.1)

Description.

127. The purpose of the performance measures information file is to provide examples of measures and to discuss considerations for measure selection.

Content.

128. Determination of what to measure and what types of measures to use is an important aspect throughout system analysis, design, and evaluation.

129. Meister (1985) suggests the following types of behavioural measures of performance:

a) Time to complete an action or produce a result
b) Time sharing among events and among tasks
c) Accuracy of response
d) Frequency
e) Amount achieved
f) Resources used

130. Meister indicates the following criteria for selecting measures:

a) Data availability
b) Data reliability
c) Relevance
d) Sensitivity
e) Objectivity
f) Suitability
g) Generality
h) Comprehensibility
i) Utility

131. Lane (1986) added the following considerations for the development of good performance measures:

a) The purpose of measurement and the information goals must be established.
b) Utility relates to the practicality and economy of obtaining the measurement information.
c) Credibility means that the performance information is believable.
d) Sensitivity is related to the information need. Complex situations pose the problem of variability that reduces the sensitivity, so real differences cannot be measured.
e) Separability means that the important components influencing outcome measures can be disassociated and separate causes can be identified.
f) Comprehensiveness means that nothing important contributing to the performance of interest is left out.
g) Specificity relates to the focus of the measure on a particular aspect of performance. Individual performance factors can be recognised and quantified so sources of problems can be diagnosed and selective improvements made.

132. Garlinger and Fallesen (1988) assessed the strengths and weaknesses of nine measurement techniques used for C2 training on ten criteria. The criteria included desired qualities of measures:

a) Data should be available for timely feedback.
b) Should provide diagnostic capability.
c) Should discriminate among true differences.
d) Should be reliable and consistent.
e) Should have high construct and face validity.
f) Should be easily administered.
g) Should be easily scored.
h) Should be accurate.
i) Should be objective, not unduly influenced by individual opinion.
j) Should have potential to be automated.

133. They identified five sources of C2 information that can be measured and three classes of

measurement techniques. The information sources were C2 products, procedures, human knowledge, decisions, and results. The classes of measurement were observation, testing, and statistics. Observation involves some degree of human scrutiny over performance. The Headquarters Effectiveness Assessment Technique (HEAT) and the Army Command and Control Evaluation System (ACCES) are two fairly comprehensive protocols for observing command post performance. Testing places the requirement on those measured to engage in or demonstrate some behaviour or knowledge. Statistical measures yield data related to mission or scenario outcomes. For example, statistics are derived from the computation of battle statistics like loss-exchange ratios. The assessments are summarised in Table 5.1. The following was concluded:

a) External observation should be preferred over peer or self assessments.
b) Probes can present situations for measurement of specific performance.
c) Testing techniques, like information flow, are better than observation or summarisation techniques on objectivity, accuracy, validity, and reliability criteria.
d) Multiple measurement techniques should be used to diagnose cause and effect relationships.
e) Exploration of the relationships among sources of performance data (i.e., products, procedures, knowledge, decisions, and results) will lead to better measures.

Table 5.1. Sample Comparison of C2 Measures (adapted from Garlinger and Fallesen, 1988)

Category	Technique	Strengths	Weaknesses
Observation	Self assessments Peer assessments	Timely feedback Reliable Ease of scoring	Not objective Questionable accuracy
	External observers (using HEAT protocol)	Discriminates (100 point scales)	Complicated to administer and score. Questionable objectivity. Unknown reliability, validity, and accuracy.
	External observers (view tapes using MAPP protocol)	Reliable Ease of scoring	Low discrimination
	External evaluators (ARTEP)	Timely feedback Ease of scoring	Questionable reliability, validity, and accuracy
Testing	Probes (insert information, events, etc. to elicit behaviours)	Potential to diagnose Discriminates	Careful preparation required. No inherent, associated method of collection and measurement.
	Information flow questionnaires	Easy to score Objective Discriminates Reliable (internal consistency)	Intensive preparation. Limited to specific tasks and dependent on later memory recall.
	Comparison of tactical maps to 'ground truth'	Potentially diagnostic and objective	Feedback not timely. Collection is involved. Un-established scoring techniques. Assumes maps reflect human knowledge.
Statistical	Results (relative exchange ratio = enemy losses/ friendly losses)	Reliable Usually objective	Low diagnosticity. Questionable validity. Accuracy depends on underlying combat models.

134. Engel and Townsend (1991) recommended eight basic outcome measures for C2:

a) Planning time
b) Planning information exchange
c) Planning processes
d) Quality of plans
e) Command post information handling
f) Comprehension of tactical situation
g) Monitoring information exchange
h) Plan alterations

135. Selected examples of more specific measures (from a set of over 120) from the Army Command and Control Evaluation System (ACCES) (Crumley, 1989; Halpin, 1992) are given in Table 5.2. Other categories of measures include accuracy and the number of queries on incoming and outgoing messages; duration and frequency of coordination of information; completeness and time of course of action analysis; and the duration and clarity of directives (orders).

Table 5.2. Example measures from ACCES
(adapted from Halpin, 1992)

Measure	Definition	Supporting Measures
Plan duration	Time from implementation of the plan to time it is changed in some substantive way or completed.	Duration of each of the four major plan elements: mission, task organisation, boundary, and schedule.
Plan stability	A percentage based on time: Plan duration vs. intended life of the plan.	Stability of each of the 4 major plan elements.
Plan execution	A percentage based on the number of all major plan elements which are completed within original contingencies, indicating sufficient leeway for adaptation to battlefield conditions.	None.
Planning success	A percentage based on the number of plans which are completed without change ("dominant") or within original contingencies ("adaptive") compared to the total number of plans	None
Planning initiative	The percentage of all plans that are "proactive" or "contingent" rather than "reactive."	None
Planning cycle time	The time from awareness of need to the time directive is issued.	Planning cycle time under each of 3 conditions: low, moderate, or high "stress."
C2 impacts on plans	The percentage of changes to plans that are not attributable to the failure of C2 processes.	Impact of information handling, situation assessment, course of action, directives, information exchange, and outgoing information.
Accuracy of assessments of the situation	The percentage of assessments about forces that turned out to be correct	Number of elements considered in comprehensive or causal assessments.
Time span of assessments	The time from the expression of an assessment to the end of the period that the assessment covers.	None.

Rationale.

136. Good measures should be used throughout COADE in task analysis, modelling, performance analysis, and even when specifying requirements and evaluating them. The desired qualities that measures should have were presented (Garlinger & Fallesen, 1988; Lane, 1986; Meister, 1985) so measures can be appropriately selected, combined, and implemented.

5.1.2.2. Performance Standards (R.2)

Description.
137. Performance standards are needed to be able to assess performance of the decision maker-decision aid system and to make a judgement that a problem exists or that better performance can be achieved.

Content.
138. The purpose of having standards is to allow comparison. Having explicit standards allow little question about the interpretation of performance results. When task or cognitive performances are assessed, they would be meaningless unless they can be judged in terms of acceptance. Even when there are no external standards, analysts (as well as stakeholders) subjectively judge whether performance is acceptable or not. Explicit standards are very difficult to establish for C2 performance. C2 performance is subject to so many potential influences that it is difficult to determine what normative levels should be. But the difficulty of setting standards does not diminish their importance.

139. Whether standards are set high or low can depend on whether the purpose is to allow a larger portion of the tasks to meet or exceed the standard or whether setting higher levels will stimulate greater efforts by performers, trainers, and system developers. This discussion of standards, and COADE in general, recognises that however desirable it is for standards to be objective, they really are subject to change both in absolute level and on the basis of which measures they are established.

140. The following information provides several concepts for how standards can be derived:

a) Historical data. Historical data can be used to set a standard for performance. The average, median, or mode of data might be used to set a standard for the future. Other statistical approaches can be used like confidence intervals or entire distributions instead of single points of central tendency. The historical data can come from reference or predecessor systems and adjusted for necessary changes.

b) Mission requirements. Analysis of requirements can be done to set standards based on overall constraints of jobs, environments, or missions. For example, an overall mission must be completed in 30 minutes (based on the enemy's capability to respond), so one critical task in an overall series might be set at some fraction of 30 to allow completion of that task and the rest of the tasks. Work programming and scheduling techniques like critical path analysis and program evaluation and review techniques (PERT) can be used to determine desired times for tasks. Performance accuracy standards for individual tasks might be set by separating the overall standard into component parts. For example, if an accuracy rate of 90 percent is required, two tasks might be set at 97 and 93 percent, if they have a multiplicative relationship (similar to system reliability calculations). Other task relationships might be additive or conditional wherein simple addition and Bayes theorem could be applied.

c) User-defined. Standards might be established by having the user or sponsor set desired levels of acceptance. This is similar to the HEAT and ACCES methods where performance standards are based on each unit's standing operating procedure (SOP). In the absence of previously documented organisational standards, the analyst can work with the organisation to elicit their standard.

d) Human capability limitations. Standards can be based on performance limitations. This is similar to the use of micro level processes to model performance (Card, Moran & Newell, 1986). Findings and theories about micro-level operations (such as eye movements, simple reaction time, listening time, speaking time) can be combined to calculate a standard.

Rationale.
141. Standards are critical to assessing the adequacy of performance. Establishing standards

may be an issue of iterative testing. If the standard is a derived one (as opposed to one set by the organisation or carried over from previous systems), it may need to be tested for feasibility, to determine what proportions of performance fall above and below the accepted level.

5.1.2.3. Types of Behavioural Errors (R.3)

Description.

142. Performance analysis can benefit from the use of previously identified types and causes of errors. This reference information provides several classification schemes for errors.

Content.

143. There are several categorisations of behaviour that are useful for classifying types of errors. One general approach is to consider the types of errors that are associated with processes and knowledge structures. Erroneous performance may be described by the following types of failures:

a) Failure to perform a process (errors of omission)
b) Perform a process when it is inappropriate (errors of commission)
c) Perform in the wrong amount
d) Perform in the wrong combination
e) Perform in the wrong order

144. Different types of knowledge errors may cause the above performance errors:

a) Flawed
b) Extraneous
c) Incomplete
d) Missing
e) Over-simplified
f) Conflicting

145. There is another class of errors which is emotive in nature. Errors of this type include things like social miscues, doing what you are good at instead of what is most appropriate, and putting forth minimal effort. These affective errors are not represented in the list below but may be indirect causes for the different cognitive performance errors.

146. Miller (1971) identified a list of 19 decision making difficulties in a larger list of 25 task functions. The 19 difficulties are not exhaustive of the types of functions which are required of C2 performance, but they provide another list of common limitations (see Table 5.3).

Table 5.3. Decision Making Difficulties (Miller, 1971)

Category	Decision Making Difficulties
Input state	Input variables are incomplete or include irrelevant variables. Classification structure of input variables inappropriate. Information is absent on one or more variables. Information on various input variables arrives out of time phase. Input noise disturbs the perception of relevant signals. The meaning of the situation is not adequately interpreted.
Goal variables, priorities	Goal variables are inadequately defined. Incompatible priorities exist among goal variables.
Response alternatives	Set of alternatives recognised is inadequate. Definition and classification of alternatives is inadequate. Premises for combining or compromising alternatives are inappropriate. Data on consequences of response alternatives in the specific situation are inadequate.
Strategy rules	Processor is unable to identify and select appropriate strategy rule. Strategy rules conflict. No strategy rule is available for combination of situation and identified alternatives.
Decision processor	Short-term memory (buffer) is insufficient. Logical capability to process all the data is inadequate.
Response effector	No channel exists for transmitting or executing the chosen response. No appropriate message code for converting output response into control behaviour.

147. Altman (1967) developed five categories of psychological factors to identify "molar-level" error behaviours. Two of the categories are applicable to decision making or reasoning (see Table 5.4).

Table 5.4. Example Error Behaviours (adapted from Altman, 1967)

Category	Error
Logical manipulation, rule using, decision making	Incorrect value weighting of responses to contingency. Failing to apply an available rule. Applying correct, but inappropriate rule. Applying fallacious rule. Failing to identify all reasonable alternatives. Making unnecessary or premature decisions. Delaying decision beyond the time required.
Problem Solving	Formulating erroneous rules or guiding principles. Failing to use available information to derive needed solution. Accepting inadequate solution as final.

148. Rasmussen (1982) developed a taxonomy for description and analysis of the events involved in human malfunction. He distinguished the external cause of malfunctions from the internal mechanisms and malfunctions and identified the external manifestation of the failure. He also provided analytical guides for identifying the components. Rouse and Rouse (1983) developed another classification scheme of errors based on six general stages of processing (see Table 5.5. (Reason, 1990) adopted three levels of processing hypothesised by Rasmussen and generated a generic error modelling system (GEMS). Table 5.6 describes examples of four types of errors based on the skill-, rule-, and knowledge-base levels.

Table 5.5. Human Error Classification Scheme (Rouse & Rouse, 1983)

General Category	Specific Category	Description
Observation of system state	Excessive Misinterpreted Incorrect Incomplete Inappropriate Lack	Improper rechecking of correct readings of appropriate state variables. Erroneous interpretation of correct readings of appropriate state variables. Incorrect readings of appropriate state variables. Failure to observe sufficient number of appropriate state variables. Observations of inappropriate state variables. Failure to observe any state variables.
Choice of hypothesis	Inconsistent with observations Consistent but very unlikely Consistent but very costly Functionally irrelevant	Could not cause particular values of state variables observed. Could cause values observed but much more likely causes should be considered first. Could cause values observed but very costly (in time or money) place to start. Does not functionally relate to state variables observed.
Testing of hypothesis	Incomplete False acceptance of wrong hypothesis False rejection of correct hypothesis Lack	Stopped before reaching a conclusion. Reached wrong conclusion. Considered and discarded correct conclusion. Hypothesis not tested.
Choice of goal	Incomplete Incorrect Unnecessary Lack	Insufficient specification of goal. Choice of counter-productive goal. Choice of non-productive goal. Goal not chosen.
Choice of procedure	Incomplete Incorrect Unnecessary Lack	Choice would not fully achieve goal. Choice would achieve correct goal. Choice unnecessary for achieving goal. Procedure not chosen.

Execution of procedure	Step omitted Step repeated Step added Steps out of sequence Inappropriate timing Incorrect discrete position Incorrect continuous range Incomplete Unrelated inappropriate action	Required step omitted. Unnecessary repetition of required step. Unnecessary step added. Required steps executed in wrong order. Step executed too early or too late. Discrete control in wrong position. Continuous control in wrong position. Continuous control in unacceptable range. Stopped before procedure complete. Unrelated inappropriate step executed.

Table 5.6. Generic Error Modelling System (GEMS) (Reason, 1991)

Error types	Description	Generic Error Modelling System	Examples of GEMS failure modes	Descriptions
		Skill-based level of behaviour		
Slips	Attention failure to store or execute an action	Inattention, omitted checks	Interruptions	Failure to make an attention check is compounded by some external event.
			Interference	Two currently active plans or action elements become entangled each trying to gain control of effectors.
			Reduced intentionality	An earlier intention becomes overlaid with other demands to do something.
			Perceptual confusions	Accepting look-alikes for an intended object.
Lapses	Memory failure to store or execute an action	Over-attention, mis-timed checks	Omissions	Attention (incorrectly) determines some sequence is further along, and omits steps.
			Repetitions	Attention (incorrectly) determines some action is lagging, so it is repeated.
			Reversal	An action sequence double backs on itself.
		Rule-based level of behaviour		
Rule-based mistakes	Deficiencies in judgmental process in specifying goals or planning how to reach them. Mis-classification of situation leads to wrong rule or incorrect recall of procedures.	Misapply good rules	Signs, countersigns, non-signs	Signs satisfy some of the conditional aspects of appropriate rule. Countersigns indicate that a more general rule is inapplicable. Non-signs do not relate to any existing rule.
			Rule strength	The more a rule chosen the stronger it becomes.
			Redundancy	Identify certain sequences or groupings of signs that tend to co-occur, favour stronger cues rather than countersigns.
			Rigidity	If rules have been successful in the past, tend to use them again even when not warranted.
		Apply bad rules	Conditional deficiencies	Certain properties are not encoded, encoded inaccurately, or exception rules may protect an erroneous general rule.
			Action deficiencies	Action component of rules can be "bad" by being wrong, clumsy, or inadvisable (wrong in special cases).

Knowledge-based level of behaviour				
Knowledge -base mistakes	Deficiencies in judgmental process, resource limitations and incomplete or incorrect knowledge.	Bounded rationality, biases	Availability Confirmation Illusory correlation Selective	Undue weight given to what comes to mind readily. Unwillingness to give up on current hypothesis. Misjudge the frequency of events occurring together. Attention given to the wrong features.
		Complex, ambiguous problem space and incomplete mental models	Focus on current Linear thinking Thematic vagabonding Encysting Diagnosis	More focus on current than past trends or future. Effects on entire system are not seen. Flitting from issue to issue. Linger over details at the cost of important ones. Difficulty in identifying symptoms and explanation.

Rationale.

149. The frameworks and descriptions of behavioural error types identified above help ensure that errors are identified and help achieve complete consideration of various types.

5.1.3. Cognitive Task Analysis

Description.

150. The purpose of the cognitive task analysis is to develop a cognitive model of the task. A cognitive model is a specification of the organisation of requisite knowledge, together with hypothesised mental operations, mental models, and meta-cognitive processes involved. Also it specifies how these cognitive components interact with information in the world. The model provides the basis for the cognitive assessment of the task resulting in the specification of cognitive requirements. The model is central for the further development of the system.

151. The analysis activity requires:

a) A set of concepts for a description of the task(s) in cognitive terms; the background information in Section 6.2 and Appendix A provides concepts and a general model that can be used in developing the cognitive model.

b) Methods to acquire the knowledge and processes involved; the Sections 5.1.3.1 and 6.3.4.5. provide an overview of knowledge elicitation techniques.

Products.

152. The product of the cognitive task analysis is a task-specific cognitive model including hypotheses about cognitive performance.

Methods.

153. Redding (1989) notes that methodologies for cognitive task analysis are limited and fairly ad hoc. A methodology should guide what the steps are in the analysis and where to look for, and how to acquire the information about the cognitive structures and processes involved.

154. For decision aiding the cognitive task analysis should identify the cognitive processes that are relevant for performance of the task. The cognitive task analysis is based on the following steps:

a) Identify the task behaviours to be analysed:
guided by:
- Existing literature on psychological limitations and capacities (e.g., Card, Moran & Newell, 1983; Zachary, 1988); see also Performance Aiding Strategies (Sections 5.2.1.2. and 6.5);
- Emerging literature on generic tasks and cognitive strategies (e.g., Schaafstal & Schraagen , 1992);
and focusing on:
- Tasks that are described by higher level cognitive action verbs (e.g., make decisions, diagnose, solve problems, judge, set goals, etc.)
- Tasks in which there are abnormalities in information transformation (e.g. unexpected outputs)
- Tasks in which there are performance errors as indicated from Performance Analysis.

b) Conduct initial knowledge elicitation to acquire an overview of more general information concerning principles, values, goals, schemas, concepts, categories, rules, strategies, relationships, mental models, and meta-cognitive processes involved.

c) Develop a model of hypothesised cognitive processes, knowledge structures, relevant input cues, and output products.

d) Refine the hypotheses or model.

e) Confirm or disconfirm the hypothesised or modelled cognitive behaviours.

155. Klein (1993) distinguishes four categories of methods for cognitive task analysis:

a) Questionnaire and interview: The questions are centred around a list of cognitive probes related to the focal point mentioned above, e.g. 'what information did you use in making this decision, and how was it obtained?'.

b) Controlled observation: Nonroutine incidents studied by assessing performance during the task, e.g. with think-aloud protocols, cued retrospective reports.

c) Critical incidents: Incidents in the past that required nonroutine activities are studied because they show the special abilities that are available in the task and are mostly well remembered.

d) Analytical: Methods used are the Brunswick lens model, multidimensional scaling, repertory grid.

156. Klein (1993) focuses in his Naturalistic Decision Making framework on:
a) the key decisions
b) the cues that enter into the decision
c) distinctions between cues that appear similar
d) the types of inferences involved
e) strategies for making these inferences
f) contextual factors that affect the inferences and decisions
g) categories used to classify situations
h) sources of confusion
i) types of knowledge gained through experience.

157. Two methods of the critical incident technique are Conflict Resolution and Critical Decision Method. In Conflict Resolution an incident is described by the decision maker and at different points in the episode the decision maker is asked to imagine what might have happened if the particular outcome had not occurred. In the Critical Decision Method a precise account is given of the decision steps of a nonroutine event using the probes mentioned. Subsequently, the steps are assessed from a novice perspective: where might a novice have misinterpreted events, missed cues, made wrong choices, etc.

158. Hopson and Zachary (1982) describe another methodology for analysing the decision problem, which they call the summary tabulation of aiding requirements (STAR) (also Zachary, 1986; Zachary, 1988; Zaklad, Bulger, Glenn, Zachary & Hicinbothom, 1986). The STAR involves determining information about the following:
a) Decision situation
b) Task dynamics
c) Situational objective
d) Value criteria
e) Underlying process
f) Information environment (input, output, parameters)
g) Intermediate reasoning and analysis steps
h) Representation
i) Required judgements.

159. Knowledge acquisition and cognitive modelling techniques (see Sections 5.1.3.1 and 6.3.4 on Knowledge Elicitation); offer good guidance on collecting and organising information at a deeper level than is afforded by typical task analysis.

160. Previous behavioural and cognitive studies provide examples of analyses which may not be referred to as cognitive analysis, but still provide sources on information on cognitive processes, abilities, and limitations. Amalberti and Deblon (1990) provide a good description of cognitive modelling for aiding fighter pilot process control. For C2 tasks, numerous studies exist which provide background and insight into cognitive performance. For example, areas addressed in the literature include situation assessment (Endsley, 1988; Noble, Grosz & Boehm-Davis, 1987; Thompson, Hopf-Weichel & Geiselman, 1984); option generation (MacMillan, Entin & Lentz, 1988), planning behaviours (Fallesen, 1993; McCann, 1990; McCann & Essens, 1991; Powell & Schmidt, 1988), naturalistic decision processes (Thordsen, Galushka, Klein, Young & Brezovic, 1989), and group problem solving (Lussier, 1991).

Relationship to other activities.
161. Cognitive task analysis is a natural follow-on to task analysis and provides cognitive model information for a cognitive performance analysis (5.1.4). The knowledge elicitation reference information (R.4) describes various techniques to collect information and to analyse it.

Rationale.
162. Identifying the cognitive nature of the task is of fundamental importance for analysis of the reasons for existing performance-based limitations, and for enhancing performance.

5.1.3.1. Knowledge Elicitation (R.4)

Description.

163. Knowledge elicitation is a set of techniques to tap cognitive processes, structures, strategies, concepts, and facts.

Content.

164. Distinctions among knowledge acquisition techniques, suggested by Geiwitz et al. (1990), include type of knowledge unit elicited, top-down vs. bottom-up processing, convergent vs. divergent vs. transformational, procedural vs. declarative, and case data vs. domain-specific rules. The categorisation concerns how the data are collected and how the data can be analysed. Practical guidance in performing knowledge elicitation is provided by Firley and Hellens (1991).

165. Data collection and analysis techniques for knowledge elicitation include the following (for complete information see Section 6.3.4.5 -Table 6.1):

ACT-based representation
Augmented Conceptual Ranking
Backward Thinking
Card Sorting
Cloze experiments
Cognitive structure analysis
Cognosis
Concept Mapping
Constrained Processing
Critical Decision Method
Data Flow Modelling
Decision Graph
Decision rule elicitation
Discourse Analysis
Document Analysis
Elicitation
Entity Life Cycle Modelling
Entity-Relationship Modelling
Familiar and novel situations
Free Generation
Goal Directed Analysis
IDEF Modelling
Integrated knowledge structures
Laddered Grids
Lens model
Model-Based Reasoning
Object Oriented Modelling
Observation (induction)
Petri Nets
Picture Probes
Policy Capturing (Ratings)
Process tracing
Protocol Analysis
Proximity analysis
Questionnaire
Ranking
Repertory Grid
Schema Based Knowledge
Semantic Nets
Shellsort
Static simulation
Storyboarding
Structured Interviews
Task Action Mapping
Teachback
Unshuffle

Rationale.

166. Knowledge acquisition and elicitation are very important for creating an understanding of the cognitive requirements of the decision maker. Uses and advantages of the respective techniques are discussed by Geiwitz et al. (Geiwitz, Kornell & McCloskey, 1990), and Leddo et al. (Leddo, Cohen, O'Connor, Bresnick & Marvin, 1991).

5.1.4. Cognitive Performance Analysis

Description.

167. Cognitive performance analysis involves the examination of the cognitive model of processes and structures to identify what aspects of cognition are critical for successful performance. Cognitive performance analysis should identify how cognitive performance is limited, what aspects are most frequently limited, and which are most critical. This analysis should also begin to reveal whether it is feasible to remedy the cognitive limitations and how. Cognitive limitations include errors with a cognitive component.

Products.

168. The results of the cognitive performance analysis is the specification of cognitive requirements. Cognitive requirements are statements indicating the most critical cognitive processes, structures, and controlling strategies for a task.

Methods.

169. Cognitive performance analysis has no methods established outside of COADE. The suggested methods for identifying cognitive limitations and specifying cognitive requirements involve

a) a deep and traceable understanding of the task (Section 5.1.1)
b) how well the task is performed (Section 5.1.2)
c) what the cognitive components are for critical subtasks (Section 5.1.3)
d) how the processes, structures, and controlling mechanisms of cognition may be suboptimal (This Section).

170. In general terms, the first stage in this approach is a good understanding. COADE recommends successive level of detailed consideration, from system and task level to observable performance behaviours to the underlying cognition for the task. The analyst's understanding is developed through identification of characteristics of the task, the decision makers, the organisation, and, performance behaviours. More specifically, a good understanding is developed from knowledge elicitation about what knowledge is used and how it is used. Indications of cognitive limitations are developed during the cognitive task analysis. The building of a model of cognitive task performance provides an explicit representation of task-specific cognition.

171. The second stage in this approach involves hypothesising about what the cognitive limitations might be. A list of cognitive limitations is provided (Section 5.1.4.1) to assist in the exploration of potential errors. The limitations are based on findings or principles about how cognition operates. The list identifies limitations from a schema-based approach to cognition, a learning approach, and a component element approach. The latter way identifies limits associated with mental knowledge representation, reasoning processes, and controlling processes (meta-cognition). The intent of the list (Section 5.1.4.1) is to prompt a range of viewpoints to explain the cognitive limitations.

172. The third stage in this approach is the statement of cognitive requirements. The primary effort here requires the determination of which possible aspects of cognition and which limitations (of which there will probably be many) should be termed as a requirement. There are four criteria to help with the selection:

a) <u>Criticality</u>. Limitations are critical if they are primarily responsible for performance errors. If limitations are critical then they should be leading candidates for remediation and statements of cognitive requirements. If criticality is unknown for a specific situation then the remaining criteria should be addressed.
b) <u>Frequency</u>. Attention should also be given to limitations that occur frequently, even if their effects are not critical. Supporting performance by reducing frequent errors can lead to lower demands on cognition and maintain greater cognitive resources for critical aspects.
c) <u>Self-correction</u>. When certain limitations are critical or frequent or both, it should be determined whether decision makers impose their own corrections. Indicators for self-correction include whether or not the decision maker looks for errors and recognises them, and how well errors are corrected when they are identified.

d) Feasibility of correction. If it is not clear how to correct a frequent or critical limitation, the analyst should still specify a cognitive requirement. If other critical or frequent limitations present a feasible solution, then these should be considered (at least) initially as the more important.

173. Cognitive performance analysis is primarily a judgmental activity. The list of cognitive limitations and the above criteria for selection of key limitations provide a general framework. There is no substitute for deep, insightful analysis when proposing and testing cognitive requirements.

Relationship to other activities.

174. Cognitive limitations are derived from the basis of information produced in the other analysis activities. Understanding of cognitive principles and decision making theories along with knowledge elicitation and evaluation activities provide the source information for determining cognitive limitations. The cognitive requirements in turn produce the basis for design requirements and should be used as evaluation standards.

Rationale.

175. The identification of cognitive limitations is necessary in order to specify a set of cognitive requirements for a decision aid. Cognitive requirements are the essence of the COADE framework and are the key ingredient left out of decision aids.

5.1.4.1. Cognitive Limitations (R.5)

Description.

176. Cognitive limitations are obstacles that get in the way of better performance. This list of cognitive limitations is provided to help ensure completeness when considering what can be done to overcome errors or to take advantage of relative strengths.

Content.

177. Table 5.7 provides an overview of cognitive limitations organised according to the set of cognitive concepts used in COADE. An comprehensive description of Table 5.7 is provided in Section 6.4.3.

Table 5.7. Cognitive Limitations

Limitations	Subcategories	Examples
Schema	Instantiation	Incorrect fitting of data to the schema. Incorrect filling of slots with guesses instead of data (similar to failing to search or construct a relevant model). Inappropriate use of subordinate schema (over-specialisation). Inappropriate use of superior schema (over-generalisation). Error in accretion: an experience is incorrectly assessed as another. Error in tuning: incorrect elaboration and refinement of concepts. Inappropriate use of most common schema: forced to fit the situation (similar to a "habit" bias). Several schemata are triggered, but wrong one is picked. Schemata are confused. Common elements represented at a higher level incorrectly called into script. Existing schemata are relied upon too heavily, reluctance to generate a specialised schema.
	Formulation	Inappropriate conditions embedded in the declarative part of the schema. Inappropriate rules or responses embedded in the procedural part of the schema. Key parts of rule conditions are omitted. (Compounding is a process of developing a new rule through simplification by intersection of conditions of two rules.) Non-standard elements of schemata have not been stored as pointers or tags, so are unavailable to form different schemata or to differentiate among existing ones. Schemata are not formulated when appropriate.
	Execution	The correct schema is activated, but an error occurs in procedure (e.g., computational error).
Learning	Classification	Links among concepts are not made or made incorrectly. Important attributes are left out of classifications when they are formed. Memory structures are excessively reorganised when new experiences and repetitions occur.
	Feedback	Insensitive to feedback (related to feedback biases).
	Rules	Rules or regularities are not generalised to induce new rules.
Knowledge Representation		Rudimentary or insufficient knowledge and relationships, including poor goals, values, or "world" knowledge. Poor organisation of knowledge. Inability to use or retrieve appropriate representations. Inappropriate crossed memories. Poor integration of knowledge or poor representation for a particular state.
Basic Processing	Understanding	Poor encoding or representation of the situation and its meaning. Ignoring important classification attributes. Failure to recognise salient features and critical relationships in a problem. Failure to consider implications of models identified in the search.
	Generalisation	Missing or inappropriate abstractions or chunking. Incorrect normalisation - transformation to an event that was not most likely or typical. Thinking occurs at wrong level of abstraction. Too few abstractions are used, too few multiple relations.

	Reasoning	Inappropriate strategy selection (incorrect schema). Inconsistent application of a strategy. Inappropriate relational or logical reasoning. Inability to hold in mind various possibilities. Poor trade-offs about importance. Ignoring uncertainty rather than trying to resolve it. Failure to project ahead. Inadequate search for counterexamples. Inappropriate use of analogical reasoning. Failure to critique, check for consistency, validity of assumptions. Failure to de-conflict ambiguous information.
Meta-cognition	External monitoring	Failure to recognise that a situation requires something to be done. Failure to gauge difficulty of a problem.
	Internal monitoring	Failure to assess likelihood of knowing. Failure to monitor actions, evaluate one's strategy. Failure to organise thoughts.
	Regulation	Inability to allocate attention and cognitive resources. Poor use of time. Failure to set goals. Inability to synchronise processes. Inability to control actions, revise one's strategy. Planning is overly opportunistic; lacks adequate control.

Rationale.

178. Definitions, examples, and descriptions of cognitive limitations, errors, or requirements are not very common in this field. The list of cognitive limitations is an attempt to provide a systematic set of possible limitations, similar to taxonomies of behavioural errors (see Section 5.1.2.3). This list offers a cognitive level category and example descriptions of various limitations. (The limitations are discussed further in Section 6.4.3.)

5.2. DESIGN

5.2. DESIGN
- 5.2.1. Solution Approach
  - 5.2.1.1. Selection and Design of Aiding Concept
  - 5.2.1.2. Performance Aiding Strategies and Guidelines (R.6)
- 5.2.2. Human-Computer Interface Design Activity
  - 5.2.2.1. Selected Design Principles, Guidelines, and Standards (R.7)
  - 5.2.2.2. Design Methodologies and Tools (R.8)
  - 5.2.2.3. Models and Architectures (R.9)
  - 5.2.2.4. Information Exchange (R.10)
  - 5.2.2.5. Promising Interaction Techniques and Technology (R.11)

5.2.1. Solution Approach

Description.

179. The purpose of this design activity is to determine the best approach for addressing identified problems and satisfying the cognitive requirements. Solution approaches other than decision aiding include changing:

a) doctrine
b) task procedure
c) function allocation
d) user dialogue
e) personnel
f) training
g) manpower
h) organisation

180. If potential approaches other than decision aiding are identified, the analyst-designer-evaluator can use the COADE framework to guide its development.

181. A secondary purpose of this activity is to help ensure that a full range of system issues and constraints are taken into account. By considering the influences of these solution approaches on an aid, special requirements can be determined, and in turn the impact that the aid will have back on the system can be assessed. It is likely that the requirements can be addressed best by a mixture of solution approaches.

Products.

182. There are four products that should result from this analysis:

a) A decision whether an aiding approach, some other approach, or a mixed approach is most appropriate.
b) Specification of an updated mission and operational concept for the solution approach.
c) Assessment and identification of changes that introducing the solution will have on the other elements of the system.
d) Identification of the human and cognitive impacts on other system requirements (not tied directly to the presentation and representation of the aid, e.g., training, data base initialisation, maintenance).

Methods.

183. Methods can be organised (see below) according to

(1) Solution approach and mission requirements, or
(2) System impacts and requirements.

184. (1) Solution Approach and Mission Requirements. Methods for determining the most appropriate solution approach fall into the general class of feasibility and cost-benefit-risk assessments and comparisons. Candidate solutions should be conceptualised. If possible, it is desirable to have different candidate approaches. The more uncertainty about the real cognitive requirements or the more the uncertainty about the adequacy of a given solution approach, the more that the approaches should differ.

185. To conceptualise candidate solution approaches, the concept of the range of missions should be projected. Mission requirements were likely to have been available in the analysis phase of COADE, and, if available, can be updated based on the different capabilities that the candidate solution would offer. The more complete and explicit the mission requirements, the easier it should be to realise the implications for the operational environment. Characteristics from the task and supporting analyses can help identify the context in which the solution would fit (also see the comments below on the impact of the aid on the system).

186. While approaches are being considered, they should be assessed briefly in terms of their feasibility for meeting the cognitive requirements. (Although quick assessments of feasibility and adequacy are implied, solution concepts might not be able to be comfortably assessed until somewhat

formal evaluations can be conducted. The candidate solution may need to be represented in some form, such as a procedural flowchart, draft program of instruction, or operational software prototype. Often cognitive requirements evolve through cycles of problem analysis, consideration of solutions, and evaluation, as part of a rapid prototyping effort.)

187. Once approaches have been conceptualised they can be compared in terms of required resources, projected benefits, and potential risks. The considerations can be quick, qualitative assessments, depending on the organisational demands and costs involved. If the organisational climate is such that change is not desired or if one solution approach (e.g., the application of artificial intelligence) has prompted the analysis and design effort from the beginning, then the search for alternative solutions and comparison of trade-offs should be more deliberate. It is when the solution approach is taken for granted from the beginning that the greatest oversights will occur.

188. The assessment of approaches might include considerations of the following factors:

a) Effectiveness of approach;
b) Development expense;
c) Ease of implementation;
d) Ability to maintain and support solutions;
e) Constraints imposed by sponsor;
f) Amount of available time until implementation;
g) Total system changes (see below: System Impacts and Requirements).

189. A generic comparison of aiding and alternative solutions is presented in Table 5.8.

Table 5.8. Examples of Trade-offs between Aiding and Alternative Solutions

Alternative Solution Approach	When the alternative is preferred.	When aiding is preferred.
Doctrine	Doctrine can overcome problems by changing mission requirements and the philosophy of how they are addressed. Problem performance can be avoided by modifying doctrine to do away with the performance requirement.	When changing doctrine is undesirable or not warranted, aiding can offer approaches that are more technical, offer more knowledge, or help ensure current doctrine is adhered to more closely.
Task procedures	Procedural changes may be lowest cost of alternative solutions.	New procedures may not be implemented as intended (not understood, forgotten, used by some-not by others, etc.). Procedures embedded in an aid are "self-enforcing" (if the aid is used).
Function allocation	There is some chance that equipment may have existing capabilities to take on additional functions to prevent problems, (e.g., increase a computer storage buffer so more incoming messages can be handled).	Considerable equipment redesign is likely for most cognitively-induced problems. Software aids (or other alternative solutions) would generally be indicated.
User dialogue	Dialogue improvements can provide simple (low risk) but feasible solutions like improving attention, access to information, etc.	Changes in a dialogue's presentation alone may not address underlying cognitive problems. Aiding offers procedural assistance for using information available through the computer dialogue.
Organisation	Workload can be balanced or functional assignments can be prioritised and given to the most experienced personnel.	Just re-organising personnel may not resolve underlying problems. None of the personnel may have the capability to perform the function. Aiding can offer the capability for very technical or intricate solution procedures.
Training	Trained personnel are usually more flexible to meet changing situations than an aid. Training is usually more easily modified than changing an aid. Training generally has lower development costs.	Once an aid is developed its costs are extremely low compared to the recurring time and costs involved in training.
Personnel	Skilled personnel are usually more flexible than an aid.	Aid performs consistently. Could reduce errors induced by stress.

Manpower	Additional experts may be used when a problem needs to be resolved so quickly that there is not enough time to develop an aid. Additional manpower has secondary benefits of addressing new tasks, offering greater flexibility in organising for problem solving.	A well-developed aid can reduce manpower requirements.

190. Predecessor and reference systems are valuable sources of information for approach determination, as well as more specific, design issues.

191. (2) System Impacts and Requirements. Analysis of system changes involves the consideration of the decision aid (or other solution approach) in the larger context of the system. Task analysis and related methods can assist in this endeavour (see Section 5.2.1). Projecting the implementation, training, manning, sustained use, maintenance, and support of the aid into the system context will allow the assessment of what effects the aid will have and what accommodations have to be made in the system. The evaluator should try to think of unusual ways that the aid might change the current system and non-obvious interactions that the aid may have with other system components. For example one might address how the following would be different: manpower, skill levels, training, organisations, procedures, and equipment.

a) Manpower. It is not unusual for aids to cause an increase in workload and increase manpower requirements. Computer aids tend to be formal, structured, and analytical. Deliberate analysis takes more time and effort than intuition. Even when the computer performs the calculations, data is often put in manually.
b) Personnel. Sophisticated decision aids (which emulate experts) may cause a demand for an increase in the skill level of the user or operator or demand special skills (e.g., the understanding of multiple regressions). There may be concerns with user acceptance. Potential users may perceive a change in their roles as undesirable. (See Mackie & Wylie, 1985; Morris, 1987; Riedel, 1988, for ways to increase user acceptance.)
c) Training. Any new device requires new knowledge to be able to use it and operate it. If higher level skills are required, more training or more sophisticated training will be required. Manpower increases may require more training devices and more training time to be scheduled. Tasks may be changed, so new and different training programs and materials would be required.
d) Organisation. An aid may require information from other organisational elements or there may be similar requirements but the aid requires more precision, more exact formats, and more rigid time tables for input. Additional organisational elements may be consumers or users of an aid's result. This may influence how the organisation is structured (location, communication channels, lines of authority). Other elements may have a need to understand about the aid (what its bounds are). New organisational elements may be required to set-up and maintain data bases, knowledge bases, and the hardware.
e) Tasks and procedures. Aids can change the nature of the jobs and tasks. New task requirements can be projected by repeating task and performance analyses by envisioning the aid as a normal part of the operation. As indicated above in Manpower, aids can impose greater levels of structure and greater demands for precision and more information. The applicability of the aid and the limits of its use may not be apparent to the user. Additional procedures may be required to test the applicability of the aid for specific instances when they are encountered (e.g., a checklist to follow for conditions and assumptions).
f) Equipment. Equipment requirements can increase. If satisfactory host computers are not present before the implementation of an aid, they will be needed to enable the use of the aid. If equipment is present for basic information processing, the addition of aiding functions can require more devices for more users or as backups.

192. Other changes can occur in standards, performance, group process, functional allocation, information, and goals. There may be attitudinal and social impacts from introducing an aid as well. Implied system changes should be considered with other members of the system development team. The more radical the change the more effort needed for anticipating, verifying, and precluding potential problems. System changes should be a continual consideration throughout the design process.

Relationship to other activities.

193. The determination of a solution approach should be done after the problem and cognitive requirements are well-understood and before much effort is invested in selecting an aiding strategy and its design. This will not always be possible. Several approaches may have to be tried and evaluated to satisfy requirements or even to refine the level of detail of the requirements. The specification of the mission concept is important for both the selection of the solution approach and the more detailed design. Consideration of system changes can benefit from repeating the task and supporting analyses, but in a projective manner with the aid included. This activity primarily impacts the activity of aiding concept design. But also multiple solutions, mission requirements, and potential system impacts will influence the identification of evaluation issues and data requirements.

Rationale.

194. The selection of a solution approach should be done after the problem is well-understood and the cognitive requirements are identified and before too much effort is invested in selecting an aiding strategy and its design. Since an aid can change the work process, careful steps should also be taken to determine the effect that the aid will have on the rest of the system. Since the aid is likely to be principally a software product, it may not go through the normal development considerations for major hardware items. Early consideration of how the aid might change the system, should allow sufficient lead time for addressing system changes and refining the design.

Example.

195. In one aid development, the aim was to provide automated assistance to an intelligence analyst. Without an aid, the analyst would receive a constant stream of messages from various intelligence sources. The task required the analyst to identify and classify electronic emitters on the battlefield. An aid was developed that closely mimicked the manual technique used by the analysts. The aid used production rules that considered each message, identified and classified the emitter, and then made recommendations to the operator. It used textual output and had backward chaining capabilities so it could provide explanations for its recommendations.

196. As the analysts began to use the aid, they realised that their roles had changed. Whereas before the analysts were content to classify and identify individual emitters (now something the aid helped them do), the freeing of their time made them also want to define the relationships among the individual emitters. Their requirement had changed from classifying individual emitters to understanding the entire electronic order of battle. In the aid's original configuration with text output, it was difficult to see the electronic order of battle or the relationships of the various emitters. A follow-on program was undertaken that dramatically changed the aid and the output to the operator. The follow-on concentrated on the "big picture", providing a graphical depiction of the geographical locations of the emitters and the linkages among them. More complete consideration of the real cognitive requirements of the task in the beginning could have saved considerable time and cost of developing two aids.

5.2.1.1. Selection and Design of Aiding Concept

Description.

197. The purpose of this activity is to select aiding strategies and develop design specifications based on the cognitive requirements.

Products.

198. The results of this activity will be design requirements and the decision aid. Both the design requirements and decision aid will change and evolve as the analysis, development, and evaluation progress.

Methods.

199. Although COADE intends to guide the development process, it must be recognised that design is an art. Design by its very nature is creative and so any attempt to rigidly prescribe it would produce brittle, look-alike solutions. The COADE design activity is simply a set of factors for the analyst and developer to take into account in the design.

200. Reference information R.6 (Section 5.2.1.2) provides information on decision aiding strategies. There are three categories of performance aiding strategies: information handling, reasoning processes, and control of reasoning processes (meta-cognition). The description of goals and basis of aiding operations should be useful for transforming cognitive requirements into design concepts. Example labels provide additional description of the strategies. Strengths and limitations of the strategies suggest ways in which the strategies should and should not be incorporated into aids. The strategies should be used to encourage broad thinking about possible solutions. The various strategies can be modified and combined to derive a candidate aiding concept.

201. This is similar to approaches presented by Hopple (1986) and Zachary (1988) that match the type of analysis to available methods. However, the major difference is that the performance aiding strategies include tasks from more of a behavioural and cognitive perspective than the analytical approaches of Hopple and Zachary.

202. The reference information (R.6) also provides general guidelines about what to avoid in design. Ten guidelines are presented based on lessons learned from decision aids that have failed. These lessons emphasise the importance of requirements analysis, learning from past decision aids, making smart decisions about how the aid should work, and accounting for future change.

Relationship to other activities.

203. The justification of a solution approach (Section 5.2.1) should set the groundwork for exploration of design concepts. The design process will be intertwined with successive refinement of cognitive requirements and design concepts. Determination of the aiding concept will likely involve an iterative process of analysis (Section 5.1), design concepts (Section 5.2.1.1), and verification (Section 5.3.3). A suggested way of verifying concepts is to compare several different aiding approaches.

Rationale.

204. This activity makes up the essence of design. While the analyst will not usually have the sole responsibility to specify the design, the list of performance aiding strategies allows the analyst to have a positive impact on the design team. The list of strategies, organised by cognitive processes and performance goals, can be used to derive the basis of support. The analyst can also encourage the design team to avoid the pitfalls of past failures by reminding them to pay attention to traceable requirements, functionality, manageable scope, previous experience, user acceptance, flexibility, accuracy of data relationships, false precision, training, and the context in which the aid will be introduced.

5.2.1.2. Performance Aiding Strategies and Guidelines (R.6)

Description.

205. Types of performance aiding strategies can be organised into three categories: Control of reasoning processes, Reasoning processes, Information handling. The analyst and design team can use this information to consider various ways in which the aid could work. The list presented in Table 5.9 is not meant to be exhaustive but illustrative of how design can be linked to cognitive processes and performance goals. In addition ten guidelines for aiding are presented. These guidelines indicate how problems from past decision aid failures can be avoided.

Content.

206. A Taxonomy of Performance Aiding Strategies (Table 5.9), and Guidelines for Aiding (Table 5.10).

Rationale.

207. Delineation of aiding strategies can help the design team develop a cognitively-centred concept that otherwise not be considered. Guidelines for decision aid development are an appropriate means of assisting design. Other development methods have attempted to make the matching of task characteristics to an aiding strategy systematic (for example, for a visual task use graphical representation or for a choice task use decision matrices). Guidelines are less prescriptive, but are more sensitive to actual design concerns. When the guidelines are used with the taxonomy of goal-based strategies more detailed requirements can be generated.

Table 5.9. Taxonomy of Performance Aiding Strategies

"Goal"	Basis of Aiding Operation	Type Examples	Strengths	Limitations
Control of Reasoning Processes				
Attention: Perform at right time	Direct user's attention	Discriminate, Classify Filter, Alert	Focuses attention to enable better application of mental resources.	Misdirected attention may seriously degrade performance.
Quicken: Apply more quickly	Quicken	Rapid trial and error	Easily implemented in well-defined tasks.	Quicker is not always better.
Broadening: Broaden considerations	Generate additional options, criteria, attributes, outcomes Stimulate creativity	Brainstorming Decomposition Recombination	Increases search space and thoroughness.	May require expending effort with unknown probability of results. More options but not better.
Sequence: Guide appropriate sequence of performance	Alter sequence or make sequence explicit	Procedural guide Backtracking	Useful for highly procedural tasks. Backtracking is useful when possible to undo steps.	Difficult to determine universal procedures for dynamic tasks.
Arranging: Arrange efficient process	Allocation and scheduler of work resources	Pert, Gannt planning tools	Standard manual techniques and commercial tools exist.	Difficult to apply typical tools to the proper level of granularity.
Understanding: Enable understanding or encourage better one	Deliberate assessment	Procedural steps	Useful in well-understood tasks.	Inappropriate representations may seriously degrade performance.
	Change problem representation	Disrupt psychological set		
Screening: Screen knowledge	Determine and show whether prior knowledge is available.	Scanning of knowledge base. Query typical retrieval cues.	Impacts whether the decision maker needs to search for information or procedures.	Time to screen or judge may detract from solving the problem.
Difficulty: Judge problem difficulty	Assess whether strategy exists for problem or similar problems	Assess novelty of problem characteristics.		
Selection: Choice of strategy	Select appropriate strategy for the problem	Matching strategies to problem characteristics.	Useful for novices or novel problems.	May require excessive effort compared to gain; result of strategies may not differ.
Consistency: Apply consistently	Capture user-strategy and "enforce"	Bootstrapping Work completion check-off	Useful to get experts to perform consistently.	May be difficult to adapt to individual users.
Completeness: Avoid omissions	Procedural checklist or checklist of factors to consider	Principles of war METT-T	Typically simple to develop and use.	May preclude shortcuts or be too rigid for variety of situations.
Standards: Comply with standard	Observation and 'overwatch', performance assessment, feedback	Embedded performance monitoring	Could reduce personnel overhead, if properly implemented.	Difficult to develop automatic measures. Low user acceptance.
	Fault detection and diagnosis	Error checker, monitor, error remediation	Useful when faults can be reliably determined or predicted.	Faults in complex operations can be non-standard.
	Application of computer solution	Automatic take-over Lockout	Useful when a decision does not affect subsequent decisions.	Applicable tasks are not well understood. Low user acceptance.

"Goal"	Basis of Aiding Operation	Type Examples	Strengths	Limitations
Reasoning processes				
Understanding: Assess situation.	Improve understanding.	Instantiate schema	Addressing the right problem is critical.	Difficult to develop robust procedures and knowledge.
Choice: Choose or decide.	Reduce capricious, serendipitous, biased or inconsistent choices by quantitative means (systematic, repeatable, explicit numerical relationships, exhaustive evaluation, compensatory)	Quantitative comparisons, structuring of human judgements (decision analysis, MAU, SEU, Bayes, cost-benefit -risk analysis)	Formal, mathematical basis. Normative models. Provides audit trail. "Actuarial judgements are better."	Models may be too difficult to understand. Required judgements are difficult and time-consuming. Assumptions may be violated.
	Qualitative (categorical, ordinal, or noncompensatory relationships, streamlined comparison)	Elimination by aspects, Satisficing, Dominance, Maximin, Maximax	Allows quick elimination of alternatives.	May overlook important considerations. May be difficult to determine threshold values.
Planning: Formulate or search for workable plan or choice.	Calculation of "optimal" constraint and goal satisfaction.	Solve simultaneous equations, calculate relationships	Similar to quantitative choices. Rigorous. Can make use of massive amounts of data.	Similar to quantitative choices. Can be difficult to provide required data. ("Garbage in, garbage out.")
	Formulate plan, meet goals and constraints.	Qualitative planning	Imparts focus on critical task other than choice.	Difficult to construct. Limited use for complex decisions.
	Assess plan.	Plan evaluator.	Usually at least marginally successful.	Criteria may be difficult to specify.
	Allocation, scheduling, synchronisation of forces, contingency planning.	Tasking charts Synchronisation matrix Sensitivity analysis Decision tree mapping	Imparts focus on critical task other than selection.	May require considerable situation specific information.
Prediction: Predict, forecast futures	Make predictions	Qualitative predictions	Expectations provide a basis for monitoring situations and controlling actions.	Qualitative techniques can be based on uncertain or unreliable judgements.
	Extrapolate trends	Regression Time Series	Numerical trends allow repeatable predictions.	Quantitative techniques require considerable data.
Expertise: Mimic 'expert' performance	Provide experience-based knowledge	Case-based, schemata	Use when there is extensive experience in a decision situation, that can be expressed logically, explicitly. Usually an explanation capability is provided.	Can be inflexible approach when relying on canned rules. Belief structures may be contrived and artificial.
	Assist with heuristic-based judgement.	Representativeness, availability, anchoring, best guess	Allows quick judgement.	Ignores important information (base-rates, chance, illusory correlations).

"Goal"	Basis of Aiding Operation	Type Examples	Strengths	Limitations
		Information handling		
Patterns: Support comprehension of patterns	Classification according to spatial forms.	Pattern recognition. Decision support templating	Natural, isomorphic means of presenting relationships.	Possible misclassification because of insufficient attributes.
	Interpretation of meanings of patterns.	Pattern analysis.	Useful for proposing meaning of complex, unusual patterns.	May be difficult to implement.
Graphics: Represent information graphically	Graphical displays (of quantities, trends, relationships; uncertainty; probability)	Logistics Combat power Task organisation	Provides ability to see spatial and quantitative relationships.	Scaling of magnitudes can be misleading.
Maps: Use isomorphic representation of spatial information	Map (topological, geographical) displays	Overlays, coverage, performance envelopes, highlighting, selective combination and separation, visualisation, animation	Standard means of presentation. Considerable technological efforts continue.	Information is not always optimised. Information available is displayed instead of information needed.
	Map (spatial) analysis	Distance and time and rate calculation	Simple, but prone to computational errors.	May oversimplify, ignore important considerations.
Uncertainty: Mitigate effects of dealing with uncertain information	Reduce uncertainty, hedge against adverse consequences	Delay to obtain more information. Absorb uncertainty ("specious precision"). Better weighting of unreliable data, prepare for multiple possibilities. Estimate max and min values. Generalising, etc.	Can overcome obstacles of not having complete knowledge.	Dangers of erroneous handling or extension of information. Reweighting depends on accuracy of recoding low-fidelity data.
Accuracy: Increase accuracy	Calculation, spreadsheets	Technical calculations Combinations of variables	Computer generally has advantage in precision over human, reducing workload or allowing more throughput.	Amount and accuracy of input data can be a concern. Sometimes mis-applied.
Statistics: Explore and describe quantities	Quantify distributions and relationships	Descriptive and inferential statistics	Standard techniques are available.	Require rigid assumptions. Results are often difficult to interpret.
Memory: Extend memory capacity, increase access to information, quicken storage and retrieval,	Information and database management (organisation, classification, retrieval)	Relational database, Hypertext, etc.	Improving memory is useful in most tasks. Success is related to ease of retrieval.	Inflexible. Combination logics for retrieval can be difficult to use.
permit exhaustive search, increase efficiency.	Note-taker and organiser	List of planning assumptions	Simple to implement.	Not especially powerful, focus is on recording not on solving.
Documents: Improve production of documents	Production of text	Semi-automated to automated	Saves time.	Style may not capture necessary nuances of meaning.
	Checking of text	Text editor	Commercially available.	May follow overly strict grammatical rules.
	Transcription of text	Word processor	Commercially available. Users familiar with capabilities.	Only addresses routine clerical tasks.

Table 5.10. Aiding Guidelines

Requirements. **Base design requirements on analysis.**
Often decision aiding efforts begin with the notion that a technology or approach could be used for some job or task, instead of examining the task first for performance deficiencies or opportunities. The actual ways decisions are made are ignored, when a technology (like hypertext) or specific technical approach (like a normative choice method) is assumed to be desirable. The solution approach is selected well before the nature of the task, the decision makers, and the cognitive processes are sufficiently understood. Often the development or application of some technology becomes the goal, rather than improving decision making performance. Solutions cannot be force-fit, using the latest technology available or a favourite approach that a designer (or analyst) believes to be rational or normative (McCann, 1989).

Functionality. **Determining what the aid does needs to be a deliberate, justifiable process.** The selection and design of aiding strategies are complex activities. What the aid does and how it represents the context in which it helps are perhaps the most difficult aspects of effective aids. Aid proponents sometimes view aiding as automating parts of a task, providing support for what the users request, or designing enhanced information displays. While any of these approaches might lead to a workable aid, ignoring the functionality makes any candidate performance improvement only a gamble. The function or behaviour of an aid must be considered before the interface issues. What the aid intends to do, its goal, and strategy for doing it must be dealt with, even at the cost of neglecting "look and feel" issues (Fallesen, 1991).

Scope. **Develop aids that focus on specific problems.**
Several decision aiding efforts have been quite ambitious, hoping to automate all aspects of mission planning or information processing. This type of approach is based on the notion that information once stored in a computer form can be used to support multiple tasks. And once the process and results of tasks are represented, then that information can be used in subsequent tasks. Such exhaustive representation and storage may be useful if tasks are straightforward and highly procedural. C2 tasks do not usually have these characteristics. C2 involves uncertain information, goals, and procedures. The associated thinking is not easily codified in computerised form. Generally, the "grand" aid that promises to do everything does not work well (Fallesen, 1991; Nickerson & Feeher, 1975). The grand approach tends to focus too much on computer representation instead of performance.

Experience. **Avoid repeating decision aiding failures.** Decision aiding does not seem to be easy to implement for military applications. There have been numerous attempts at trying to develop decision aids for all kinds of applications. These previous efforts can provide valuable information when considering various approaches. Many aiding strategies have been ill-conceived and not particularly powerful to use in operational settings. After some aids have been developed, they are found to be fairly trivial, containing a few good rules that could have been more easily and better included in training or non-aided procedures. Analysts and designers need to identify and study previous attempts at aiding in predecessor or reference systems. These experiences should provide valuable information for why particular strategies failed or succeeded.

User acceptance. **Make explicit trade-offs between performance and user acceptance; do not simply defer to users' opinions or preferences.** User desires should be assessed and taken into account, but users should not set requirements themselves. Users' opinions should not be the only source of assessment of concepts or prototypes. Users do not necessarily have a good understanding of their performance. It is very difficult for proficient decision makers to describe how tasks are done since their abilities can be automatic. Users can be so set in the way that a task is routinely done that their suggestions may be narrowly constrained. Users may not know about alternate approaches or technologies. What users can envision may not be the same as what will improve their performance. If users have difficulty accepting effective aids, then the organisation should find ways of encouraging use (see Riedel, 1988 for user acceptance guidelines).

Flexibility. **Provide options to the user.** Build flexibility into the aid. Provide a resource "envelope," a tool box (McCann, 1989). Aids should not place barriers in procedures or concepts where they did not exist in an unaided environment (Fallesen, 1991; Weber & Coskunoglu, 1990). An aid should not be more constraining than manual procedures, unless that is the intent of the aid because some specific procedure has been found to improve performance. A desirable trait of aids should be that they are robust to accommodate a wide range of situations or to degrade gracefully over progressively more complex situations.

Accuracy. **Do not invent relationships among information.** Deterministic relationships among information are often developed to allow the use of algorithms, provide compatible comparisons, or maintain similar data structures. In one aid, an algorithm was developed for estimating resource consumption. One algorithm computed consumption based on a ratio of personnel in a subordinate unit to its parent unit, rather than using the amount of actual equipment in that unit and its projected missions (Flanagan, 1993). Although invented relationships may not be incorrect, they at least can be unfamiliar and misleading, especially in novel situations. Proposed relationships should be verified before they are incorporated into an aid. The implications of those computations need to be examined in light of other computations in the system and for future situations. Once an algorithm is embedded in an aid, it can be difficult to isolate it, trace its origins, and to correct any relationships that have been established with other algorithms.

Precision. **Do not impose a computer determined level of precision and formality on decision makers.** Computers' strong points are that they allow perfect replication and storage and error-free computations. The more that a task can be framed using exact data and rules, the more straightforward it will be for the computer. Rigidity and exactness are traits that are not strong points of human decision makers. Humans' abstraction and induction abilities are powerful for coping with complex and unfamiliar problems. Avoid false precision in aids. Do not use excessive significant digits and single measures of central tendencies. Alternatives that are more natural for humans and realistic for complex problems include fuzzy logic, confidence intervals, upper and lower bounds, comparison of worst and best possible outcomes, and sensitivity analyses (Fallesen, 1991).

Training. **Identify training requirements incurred because of the aid.** Aid proponents sometimes make the argument that their aid will reduce training and personnel. Decision makers generally need to retain at least an equal understanding of what to do in the task (even if the purpose of the aid is to inform). In fact, there is likely to be a net increase in training requirements. Besides the usual task knowledge, decision makers need to monitor the accuracy of the aid, understand the output of the aid, and know the boundary conditions of what does and does not apply. Decision makers should retain the responsibility for the task and will be the ultimate back-up if the aid cannot be used. An aiding concept that promises to obviate the need for training or reduce training costs, personnel qualifications, or manpower may be 'too good to be true.'

Context. **Anticipate the context into which an aid will be introduced.** An operational

concept needs to be developed for the environment in which the decision aid will be used. This is especially important if the decision aid is mostly driven from a technological opportunity, rather than an analysis of decision aiding needs. An aid is supposed to create change, but some of the changes may be unexpected or undesirable. An analysis of system changes should be done concurrently with analyses and design and all phases of evaluation.

5.2.2. Human-Computer Interface Design Activity

Description.

208. The purpose of this activity is to develop efficient human-computer interface designs that are well adapted to user capabilities, the tasks to be performed, and the environmental aspects which shape task performance (user, task, and environment-oriented design). Because of rapidly evolving advances in research and development in this area only a snapshot can be given here.

Products.

209. The outcome of the interface design activity will be first, a conceptual model of the decision aid interface, then requirements and prototypes in different stages, and finally the operational interface itself. The purpose of the interface is to facilitate the human-computer or user-machine dialogue through different input and output devices, hardware, and software that governs the language, style and meaning of the information interchange between user and computer.

Methods.

210. This activity is supported by descriptions of selected design principles, guidelines, and standards; by design methodologies and tools; by models and architectures; by recommendations for the information exchange design; and by a descriptions of current and future interaction techniques and technology; all with reference to different aspects of the interface design process. Citations of relevant literature regarding the overall problem space will establish the framework and the connection for these aspects of design.

211. Traditional aspects of human-computer interaction are the psycho-physical and technical design issues as related to the perceptual and motor capabilities of users, i.e. human factors or traditional ergonomics (Boff & Lincoln, 1988; Downton & Leedham, 1991).

212. Recent advances in computer technology, software development, and cognitive modelling have led to focus on cognitive aspects of human-computer interaction in decision making, problem solving, and communication tasks, i.e. cognitive ergonomics (Marshall, Nelson & Gardiner, 1987). Cognitive ergonomics, in turn, impact the methods available for the operation of the computer and control of the user-computer dialogue, i.e. software ergonomics.

213. Despite, or even because of, advances in computer technology and software development, the interface design approach must be user, task-, and environment-oriented in order to ensure an effective interface (Marcus & Van Dam, 1991; Norman & Draper, 1986).

214. The use of interface design principles, guidelines, and standards will help designers by guiding their task and will permit them to check their own work against well-established criteria.

215. Rapid prototyping is an essential step in interface design and provides a means to test and evaluate system design with respect to the requirements even at a very early phase of the development cycle (Wilson & Rosenberg, 1988). Rapid prototyping places the operator "in the loop" without requiring the technical complexity of conventional man-in the-loop simulation (Beevis & St. Denis, 1992).

216. Modelling of the user, the machine, and the interface is an appropriate and upcoming procedure in the design process. A number of different models have been proposed, each one addressing some particular aspects of interface design and/or usage (Fischer, 1990; Hartson & Hix, 1989a; Williges, 1987).

217. Models and architectures for the user interface describe the structure of user-computer interaction as a communication process performed at different levels of abstraction, e.g., presentation level, dialogue level, functional level, application level (Dzida, 1983; Green, 1985).

218. The principal function of the interface is to facilitate human-computer interaction or dialogue. To optimise the dialogue between the two partners, user and computer, the interface must accommodate the different capabilities of the unequal partners. Therefore the form of the dialogue between user and computer (style, structure, type, technique) is an important issue for task performance (Elkerton & Williges, 1989; Fischer, 1990; Marcus & Van Dam, 1991; Papazian, Roberts,

Redick, Tani, Pew & Salter, 1989).

219. User-interface management systems (UIMS) manage the dynamic behaviour of an interface, segregating the application from the interface (Hardy & Klein, 1991). The same term is often used for user-interface development systems, as well. Recent developments in human-computer interaction recommend a clear separation of application and the user interface, a concept of modularity (Hardy & Klein, 1991).

Relationship to other activities.

220. Like the overall decision aid design activity, this activity is related to Analyse and Evaluate. It must be performed in conjunction with the overall aiding system design as an evolutionary and iterative process throughout the whole system development cycle.

Rationale.

221. The rationale for a specific HCI or user-computer dialogue design activity derives from the problems people have in understanding and using computers. Recent advances in computer technology and software development offer the possibility to aid people in communicating with the computer in an interactive way using their own conceptual models of task performing.

5.2.2.1. Selected Design Principles, Guidelines, and Standards (R.7)

Description.

222. Interface principles, guidelines, and standards are the traditional medium for transferring specialist expertise to the designers.

Content.

Selected Interface Principles

223. Norman (1988) advocates user-centred design and suggests the following principles for a good design:

a) Make things visible - the user should be able to tell what is going on by merely looking at the interface;
b) Provide the user with a good conceptual model - the designer should provide the user with a mental model that is consistent and coherent;
c) Provide the user with good mappings - the user should be able to determine the relationships between actions and their results;
d) Provide the user with feedback - the user should receive full and continuous feedback about the results of all actions.

224. Usability and acceptability aspects most frequently determine successful performance and efficiency of the overall man-machine system. Because these aspects depend very strongly on the design of the HCI itself and the user-computer dialogue, as mentioned above, it is imperative that the interface be well-designed.

225. Gould and Lewis´ (1985) four principles of designing for usability capture much of what seems to be important for getting the process done properly:

a) early and continual focus on users and tasks;
b) interactive design in which potential users participate in the design;
c) empirical measurement of usage (i.e. usability testing);
d) iterative design.

226. As already suggested by Gould and Lewis, integrating user participation in the design process may offer a new potential for the purpose of achieving usability. User participation is suitable for many design tasks:

a) users can help in deciding what is needed;
b) users are valuable sources of information;
c) user´s role in evaluating the design is beneficial.

On the other hand, users' contribution to generating solutions and implementing is normally modest (Kallela, 1992).

227. The following is a short list of some principles for developing interface concepts (adopted from Galitz, 1989; Newman & Sproull, 1979):

a) Simplicity:
- the interface should not contain features that are too complex to understand;
- the interface should use conventions that are familiar to the user population;
- the interface should not be perceptually or semantically complex;
- the layout of display features should facilitate perception, understanding, and use;
- screens should not be densely filled with information;
- information should be discriminable by reinforcing user expectations with content- and style-dependent cues.

b) Consistency:
- the interface should behave in a generally predictable manner, using patterns consistent with expectations;
- identical portions of the interface should always mean the same thing and operate in the same way.

c) Adaptivity and Universality:
- the interface should be usable by the entire target population

- the interface should be flexible to accommodate individual differences among users and changing skill and knowledge.

d) Accuracy:
- the interface should be a reflection of the design and capabilities of the system
- the interface should contain a set of capabilities to effectively handle user tasks
- the interface should provide complete and prompt feedback to user interactions.

e) Robustness:
- the interface should be designed to reduce errors, and the system should promote recovery from them.

Guidelines

228. Smith and Mosier (1986) have compiled 944 guidelines in a 478 page report on designing user interface software. Deimel (1988) gives a comprehensive overall User Interface Design Guideline in the form of transparency masters from an User Interface Development Curriculum by Perlman (1989) from Carnergie Mellon University. Marshall et al. (1987) give a compilation of 162 different guidelines for the design of the user interface to complex computing systems where the emphasis is on the cognitive aspects of interface design. They place these guidelines in a framework of 14 categories or "sensitive dimensions" for interface design:

a) design of procedures and tasks
b) analogy and metaphor
c) training and practice
d) task-user match
e) feedback
f) selecting terms, wording and objects
g) consistency
h) screen design
i) organisation
j) multimodal and multimedia interaction
k) navigation
l) adaptation
m) error management
n) focus of control.

Standards

229. There are a number of standards and guidelines established by national and international authorities concerning human-computer interaction design, especially for office workplaces, e.g.:

a) ISO 9241 Part 119, "Ergonomic requirements for office work with visual display terminals (VDTs)". (ISO: International Standardization Organization)
b) CEN 29 241 (will become equivalent to ISO 9241). (CEN: Comité Européen de Normalisation/European Committee for Standardization)
c) DIN 66 234, Teil 8: Bildschirmarbeitsplätze/Display workplaces; Grundsätze der Dialoggestaltung/Principles of dialogue design. (DIN: Deutsches Institut für Normung/German Institute for Standardization)
d) VDI 500, "Office communication, Software-ergonomics in office communication". (VDI: Verein Deutscher Ingenieure/ Association of German Engineers).

230. In addition to these standards from national and international authorities (Billingsley, 1990a; Billingsley, 1990b; Billingsley, 1993) there exist a number of standards from industrial companies. Leading computer companies and software developers established so called "style guides" or "guidelines for the design of user interfaces". The purpose is to provide to the potential user a consistent "look and feel" of widely distributed computers and software. The common attributes for all these user interfaces are graphical capabilities, multi-modal input and output techniques, interaction and selection techniques, window and menu technique.

a) Apple Human Interface Guidelines. (Apple Computer, 1987)
b) IBM Common User Access (CUA). (IBM, 1991)
c) OSF/Motif Style Guide. (Open Software Foundation, 1991)
d) OPEN LOOK Graphical User Interface. (Sun Microsystems, 1990)

Rationale.

231. The use of interface design principles, guidelines, and standards will help designers by guiding their task and will permit them to check their own work against well established criteria.

5.2.2.2. Design Methodologies and Tools (R.8)

Description.

231. Various system development methodologies have evolved, corresponding to different system life cycles, such as the traditional waterfall model (Boehm, 1976), the spiral model (Boehm, 1988) and the fountain model (Henderson-Sellers & Edwards, 1990). Despite individual differences in the models and their variations, they all have components corresponding to analysis, design, and evaluation. Analysis involves problem definition and modelling. Design focuses on solution specification and modelling. System design transforms the problem representation into a solution representation.

232. There exists no single philosophy of design, but instead many different views about the nature of design. From the literature it seems to be clear that design is a complex goal-driven activity requiring tools and techniques which favour a holistic approach to problem solving. In most design situations, especially system design, the goal consists of developing a system which provides certain functions. However, functionality must also include ease of use and learning, efficiency, reliability, security, compatibility, etc. Before a system can be designed, such objectives have to be clearly specified. There are many partly different philosophies or methodologies for the interface design process that depend on the particular computer systems, tasks and situations. They generally prescribe an iterative top-down refinement process with the application of guidelines and evaluation-based backtracking.

Content.

Design Methodology

233. Hartson and Hix (1989b) propose a star configuration user interface development life cycle based on continuous evaluation. Their approach includes:

a) task analysis and functional analysis,
b) requirements specification,
c) conceptual design and formal design representation,
d) prototyping,
e) implementation.

All activities are linked together through the process of evaluation. This star life cycle, with evaluation at its centre, allows almost any ordering of development activities and promotes rapid alterations among them. Indeed, the star life cycle explicitly supports iterative refinement and rapid prototyping, thus offering the possibility of complementing the inside-out approach favoured by engineers with an outside-in approach focused on the users (St. Denis, 1990).

234. Hartson and Boehm-Davis (1993) organise their report on user interface development processes and methodologies around individual interface development activities, including brief reviews of the current state of the art and references to related work in the literature. The different sections describe design activities such as:

a) iterative design methodology
b) - system analysis
c) design of the interaction component
d) design specification and representation
e) usability specification
f) prototyping
g) formative evaluation
h) modelling

235. One of the most thorough interface design process models has been presented by Williges, Williges and Elkerton (1987) and was adapted by Papazian (1989). Their three stage model has a strong sequential component:

a) Initial Design
    Design Objectives
    Task and Function Analysis
    Focus on User
    Design Guidelines

Structured Walk Through

b) Formative Evaluation
   Rapid Prototyping
   User-Defined Interfaces
   User Acceptance Testing

c) Summative Evaluation
   Operational Software
   Benchmarking
   Formal Experimentation

236. Williges, Williges and Elkerton (1987) stress the need to have iterative generate and empirical test cycles to be able to produce adequate interface specifications.

237. Prescriptive models are idealisations of the design process that do not account for the constraints inherent in the practice of interface design. Therefore a practical description of the HCI design requirements (Brown, 1991), focusing on user participation, is desirable:

a) Designing an effective HCI requires knowing the potential users, what they are intended to do, and how the proposed decision aid will support their task.
b) The users must be involved in the design process from its earliest stages, and work with the designers to ensure the system meets their needs and is functional and usable.
c) The design must based on an understanding of the tasks the users will perform with the system and the physical and social environment in which it will be used.
d) Project-specific HCI design guidelines should be developed. The general guidelines in the literature are good starting points, but they must be tailored to the special project.
e) "Look and feel" standards may be relevant for determining the fundamental aspects of the HCI in the design alternatives. An analysis and evaluation of the specifications, advantages, and disadvantages of different approaches will be required.
f) The choice of the task-dependent dialogues and the way in which they are implemented are the most critical factor determining usability.
g) Prototypes, user testing, and redesign are three key components of the iterative design process.

Software Design Techniques

238. Several techniques have been developed to assist the software specification and design activity. Wasserman (1980) is an excellent survey in which such techniques are briefly examined. Five of these techniques are being widely used for the development of information systems and are representative of the traditional approaches to software engineering. These are:

a) Structured System Analysis. SSA incorporates four separate concepts to create a specification along with their different notations: Data Flow Diagrams (DFD), Data Dictionary, Access Analysis, and Decision Tables. Processes as well as data can be modelled and designed. The focus is on technical aspects of the system analysis but no management procedures are provided. SSA is intended to be used with Structured Design.
b) Structured Design. SD is a useful technique too support architectural design of a software product. It is concerned with the development of well-structured systems. and provides ways to compare alternative design solutions and to transform DFD into program structures. Major emphasis is given to the effectiveness of the modular decomposition where each module should carry out a single, well defined function. The major flaw of SD is its lack of support for the specification as well as detailed design and implementation phase
c) Structured Analysis and Design Technique. SADT is a hierarchical decomposition activity to produce detailed representations of a system, a process or a data structure. The modelling technique and notation is quite general, and is not specifically aimed at problem analysis into software
d) Hierarchical Input-Process-Output. HIPO was developed as a documentation aid, but has been used for the description of both specifications and design. It

consists of a collection of hierarchically organised set of modules in which each module has explicit input and output items. A module may contain either a nonprocedural description or a detailed sequence of steps that shows the actual process logic.

e) Jackson Design Method. JDM is s directed primarily at business-oriented data processing systems, but has been used successfully for control-oriented programs as well The design process essentially consists of drawing appropriate junctions between input and output data structures. The program structure is derived from data structures. In practice however, the input and output data structures seldomly fit together perfectly. JSD does not support the process of requirements analysis and specification

239. Object-oriented analysis and design (OOAD) is a new way of thinking about problems using models organised around real-world concepts. In a system decomposition based on an object-oriented approach, the system is viewed as a collection of discrete objects that incorporate both data structures and procedures in a single entity in contrast to conventional software programming in which these are only loosely connected. The objects are connected in a client-server model with messages that are sent to each other. A message invokes an object to execute a procedure. High level analysis and design is accomplished not only in terms of these objects, but also in the services they provide. Detailed design, including procedure implementation and specification of data structures, is deferred until much later in the development process. Consequently, a system based on object representation can remain more flexible since changes at the implementation level are more easily accomplished without requiring changes to the system design itself (Henderson-Sellers & Edwards, 1990).

240. Mrdalj (1990) provides a bibliography of 83 references which represent much of the research on object-oriented system development. He does not, however, provide a classification or evaluation of the research identified. Monarchi and Puhr (1991) evaluate current research on object-oriented analysis and design. Critical components in OOAD are identified and twenty-three OOAD techniques (i.e. processes or methods) and representations are compared based on those components. Strong and weak aspects of OOAD are identified and areas for future research are discussed.

User Interface Design Tools

241. Interaction techniques (e.g. dialogue box, scroll bar, pop-up menu, and text input; but also voice input or touch-screen) are the means by which users interactively communicate with the computer component in human-computer systems. The look and feel of a human-computer interface is determined largely by the collection of interaction and dialogue techniques provided for it. Designing and implementing a good set of interaction techniques is time consuming and is a critical step in bringing computer applications to end users. Using existing user interface software tools will help gaining designer productivity and ensuring a consistent look and feel among application programs (Foley, 1991).

242. Toolkits provide a library of interaction techniques that can be invoked and organised from a designer by writing the relevant code. User interface management systems (UIMS) provide additional functionality in implementing and managing user interfaces. UIMSs provide some means of defining sequences of user actions and may in addition support overall screen design, help and error messages, macro definition, undo, and user profiles. They manage the dynamic behaviour of an interface by different styles of control. Foley (1991) describes a number of graphical toolkits for designing interaction techniques, e.g. the Macintosh toolkit (Apple Computer), OSF/Motif (Open Software Foundation), and InterViews for use with X Windows. In the same paper a number of UIMSs are given, e.g. Menulay, TAE Plus, and the Smethers-Barnes Prototyper. Myers (1989) gives a good introduction to user interface development systems (UIDS) and a survey of toolkits and development systems together with a classification according to how they let the designer specify the interface. Perlman (1988) gives an overview about the usage of software tools during the user-interface development life cycle as well as for the special interface management approach. A list of 23 tools together with references is added. St. Denis (1990) gives a broad overview and detailed description of 13 systems which, to varying extents, manage specification, presentation, design, prototyping, execution, implementation, and maintenance of user interfaces to interactive systems. These system development facilities and tools incorporate important concepts regarding user interface management. According to St. Denis they have been chosen for their breadth of scope and the variety of ways in

which they approach interface management as well as the entire application system development process. Therefore, they vary widely in style and function.

243. Myers and Rosson (1992) give the results of a widely-distributed survey (74 respondents) on user interface programming. The respondents reported a significant breakdown in development time and therefore a gain in productivity using software tools. Additional subjects of the survey were topics such as computer systems and programming language used, experience of the developers, size of application, user interface of application, development process, tools used, evaluation of the tools, and most difficult aspects in developing the user interface.

Rapid Prototyping

244. A prototype is an early version of a system that exhibits some key features planned for the later operational system. Design guidelines and preliminary specifications cannot foresee all of the user-specific, task-specific or organisation-specific implications. Rapid prototyping provides a way to identify and refine poorly defined requirements so that the design problem can be detected and corrected early in the development process and at less cost. Working prototypes permit testing through operation use. They enhance understanding of the system by permitting users to experience the implications of design concepts firsthand. Also, they facilitate user-designed communications. In some cases, the final prototype iteration may become the system (Brown, 1991).

245. Basically, there are two categories of prototyping approaches: "throw-away" prototyping and "evolutionary" prototyping. A "throw-away" development process consists of building and evaluating a prototype, and then scrapping the prototype before the real system is built. A "throw-away" prototype should be built as early as possible to be most useful. This implies that the prototype normally lacks some of the intended functionality of the system and that several constraints are relaxed. The main effort would go into the critical evaluation of the prototype rather than its design and development. Clarification of requirements and system specifications is to be achieved with the "throw-away" approach. In an "evolutionary" development process, a prototype evolves through iterative modification into a complete implementation of the target application system, so that the final prototype becomes the final product. The evolutionary approach avoids the difficult question of when to discard the prototype and start working on the real system (Hartson & Hix, 1989a).

246. Metersky (1993) combines five prototype classes in a prototyping approach to decision-oriented system design and development:

a) exploratory prototypes provide the early focus for discussion during requirements elicitation and verification phases;
b) experimental prototypes provide the platform on which to evaluate the proposed software design;
c) evolutionary prototypes are concerned with the gradual adoption of operational systems to cater for newly identified or changing requirements;
d) performance prototypes are concerned with confirmation that the system will meet its stated requirements;
e) organisational prototypes are concerned with trying the system out in the end user environment.

247. Wilson and Rosenberg (1988) give an overview of issues on the integration of rapid prototyping into the user-interface design process. They describe possibilities, feasibilities, and the use of rapid prototyping, showing the components of a prototyping tool and classes of prototyping techniques. Different classes of rapid prototyping techniques are:

a) Slide show technique - a predefined sequence of display presentations; quick to assemble because it does not require detailed dialogue or logical specification; can also be done by video taping photographs, drawings and other images to produce a realistic-looking simulation of displays and devices.
b) Wizard of Oz technique - for exploring design concepts beyond the limits of available technology; a hidden human simulates the functionality of a future system.
c) Fully animated prototype - requiring considerable development effort, depending on complexity; can serve as specification of the final product.

248. Most of the software tools for user interface design are as suited for rapid prototyping

purposes. Different approaches to prototyping tools (comparable to user interface design tools) are:

a) tool kits (special procedures for design and control of graphical techniques);
b) part kits (object library with graphical component);
c) animation language metaphor (a combination of toolkits and part kits together with an animation language for dynamic simulation of an interface design).

249. A survey of rapid prototyping is given by Hartson and Smith (1991). Nickerson and Pew (1990) give a description of three prototyping tools used in industry and government for large system development as well as of tools for PCs and Macintosh computers. An evaluation of three first tools is added.

Rationale.

250. Applying existing methods, techniques, and tools can lower the burden of the human-computer interface designer and make his work more fruitful and effective.

5.2.2.3. Models and Architectures (R.9)

Description.

251. One particular method for adapting user, tasks and interface to computer capabilities is the formulation of models and their integration into the interface design and implementation process (Moran, 1981; Norman, 1981; Whitefield, 1991). Focusing on the task and the user interface allows the designer to develop a conceptual model of the system. This model makes explicit the relevant characteristics of the task, the system, and the user's knowledge of these, so that once the system is complete, the user can readily acquire a mental model similar to that of the system (Norman, 1983). User-interface design models fall into two categories: descriptive and formal. One benefit of formal models is that they could provide a rigorous way to specific system requirements (St. Denis, 1990)

Content.

User model

252. Since Norman's (1983) paper on mental models, a number of different models have been discussed, each one addressing some particular aspects of interface design and/or usage (Chignell, 1990; Hartson & Hix, 1989; Williges, 1987). The main goal is to design a system that adheres to a consistent, coherent conceptual design model, so that the user can build up his own mental model consistent with the design model (Kuhme, Hornung & Witschital, 1991).

253. A general definition of a user model for the design of adaptive user-computer interaction using a model dynamically by utilising the knowledge about the user to direct system behaviour in an interaction is given by Kass and Finn (1991):

a) user model: a knowledge source in a system that contains explicit assumptions on all aspects of the user that may be relevant to the behaviour of the system. Farooq and Diminick (1988), in a survey of formal tools and models for developing user interfaces, provide different definitions of user models dependent on different aspects or backgrounds of the researchers involved in establishing or using.
b) cognitive model (Card et al., 1983; Kieras & Polson, 1982): a model, typically formulated by a cognitive psychologist, that attempts to describe the mental processes by which humans perform some task. The usual purpose of such a model is to advance our understanding of human behaviour.
c) mental model (Borgman, 1986; Moran, 1981; Wilson & Rutherford, 1989): a model, evolving in the mind of a user, representing the structure and internal relations of a system, developed as the user is learning and interacting with the system. Mental models can be analogical, incomplete and sometimes very fragmentary with respect to their representation of how something functions.
d) user conceptual model (Mayer, 1981; Moran, 1981): a model typically formulated by a designer of a system to provide the user with an appropriate representation of that system (appropriate in the sense of being accurate, consistent, and complete).

254. Williges (1987) differentiates between the categories of mental and conceptual models and quantitative models for a user-centred approach to human-computer interface design. Conceptual models deal primarily with representation of users' cognitive processes, structures and strategies, whereas quantitative models include performance, ergonomics, computer simulation, and statistical models.

255. Whitefield (1991) examines the potential use of HCI models in the design of interactive computer systems. He gives a classification scheme and discusses the nature of system design and the role of models in the design process.

256. Following Zachary (1991) there are two general uses for user models in human-computer interface design:

a) In the first case, the user model forms the basis for critical aspects of the system design, ranging from control flow and data representation to functionality. The design approaches by Zachary (1988) and Rasmussen (1986) are detailed

examples of this approach. In both methodologies, a model of the user's representation of the system is constructed first, and then analysed to define features of the system under design and to guide critical design decisions.

b) A second and more ambitious (and hence less frequently attempted so far) use is to physically embed user models in the interface or dialogue process. With the addition of some reasoning apparatus, they can allow the interface to reason about the actions, goals, and intentions of the user and to support, enhance, and adapt the user-computer interaction to the cognitive processes of the system user.

257. There are different specific functions which an interface with an embedded user model could perform. In an intent inferencing concept, the interface infers the intent of the user and adapts the interaction as well as the information representation to that inferred intent. Rouse et al. (1987) have used an embedded user model for intent inferencing in a pilot's associate.

Dialogue interaction architectures

258. The Seeheim model (Green, 1985) was elaborated during the 1983 IFIP workshop. It was the first model that described the generic user-computer interface, having three modules:

a) Presentation component - responsible for the external presentation of the user interface;
b) Dialogue Control component - defines the syntactic structure of the dialogue between user and the application program;
c) Application Interface Model - used to represent the semantics of the application from the viewpoint of the user interface.

259. A model permitting a more detailed and specific description of the dialogue and interaction design aspects is the IFIP model (Dzida, 1983). The user interface is separated into three or four different interfaces, respectively:

a) Input output interface - definition of input and output devices and techniques, information representation, structuring, and coding.
b) Dialogue interface - definition of the dialogue between user and computer-based system, help procedures, initiative control, direct manipulative objects etc.
c) Application Tool interface - definition of user access to software tools and data
d) Organisation interface - definition of the relation to and the connection with other users and tasks, (such as, work hierarchy, cooperation).

Dialogue interaction modes

260. There are a number of different aspects related to human-computer dialogue:

a) Dialogue structure - defining the structure and architecture of the human-computer dialogue (i.e. hierarchical models, e.g. Seeheim model, IFIP model, linguistic model, implementation oriented model)
b) Dialogue type - defining the type of interaction by means of different input-output devices (e.g. graphic interaction, voice interaction, three-dimensional interaction, virtual reality, touch device, multi-media interaction)
c) Dialogue technique - defining different manipulation techniques working at the interface (e.g. menus, natural language, direct manipulation)
d) Dialogue initiative - defining the priority for starting an activity (user initiative, computer initiative, variable initiative or adaptive initiative, mixed initiative).

261. The human-computer dialogue specification approach must consider all the aspects mentioned above. A central dialogue system is required to integrate and to coordinate all these aspects for the information communication between user and interface. A dialogue controller is necessary to control information content and presentation mode at the interface (Hollnagel & Weir, 1988).

262. A thorough survey of dialogue specification techniques is given by Neilson (1987). Guidelines for dialogue design are given by Williams (1989).

Rationale.

263. Models and architectures include abstract and general approaches for defining problems in a structured manner that focuses on the essential variables and processes for problem solving. Establishing, adapting, and using models lead to better understanding a problem and the solution process.

5.2.2.4. Information Exchange (R.10)

Description.

264. There are many human factors as well as cognitive ergonomics aspects to be taken into account in designing representation of information to be exchanged between human and computer in the user-computer dialogue.

Content.

265. The three major principles governing the design of information representation at the user interface are consistency, contextual relevance, and flexibility. Partially conflicting requirements associated with these principles must be reconciled:

a) The principal of consistency requires the use of identical information presentations in terms of form and organisation; this will improve information retrieval and selection. Generic icons and standard menu and window formats can be used, and similar information should be displayed using uniform colour coding or labelling.
b) Contextual relevance means that information must refer to the given situation and the steps necessary to solve a problem or accomplish a task.
c) Flexibility means accommodating different forms of information representation and organisation and tailoring to different tasks and applications.

266. Some research findings that might assist in the design address the following issues:

a) Type of information. The type of information describes special situation or problem attributes; different types permit the situation or problem to be considered from different perspectives; therefore, different types of information should be used to facilitate "thorough penetration" of the problem (Woods, 1991).
b) Quantity of information. The quantity of information to be represented should exclusively depend on user requirements; an information overload will result in the use of only a noncontrollable portion of available information (Malhotra, 1982)
c) Form of information representation. The form of information representation or information coding has considerable influence on information transfer between the computer and the user; from the perspective of both redundancy and even utilisation of capacity, the range of human sense modalities should be used for information input and output; the superiority of graphic presentation for many tasks has been ascribed to the concept of "holistic processing" versus "sequential processing" of textual information and is further supported by research in information "chunking", i.e. cognitive integration of information (Badre, 1982); graphical presentation is done by using icons or analogue representations; shape and colour are important aspects of visual information coding; noise and pitch levels are elements of auditory coding;
d) Organisation of information. The organisation or structuring of information should ensure that information is presented at the right time, at the right place, in the right order or in the right grouping; Norman et al. (1986) describe the many different display, window, and menu options in terms of "cognitive layouts"; they caution in addition about the risk of the user "getting lost" in a multiple-display or -menu system.

267. Research findings related to human factors aspects of information representation have been summarised in the Handbook of Human Factors (Salvendy, 1987). Greenfield (1987) analyses the special impact of interaction and communication media. He underlines the fact that the data and information transformation process varies according to the medium or representation method which can emphasise certain aspects and suppress others.

Rationale.

268. Information representation for the dialogue or information exchange process between user and computer must be designed for the perceptual and, especially, cognitive capabilities of the human user. Taking results from research and adapting them to a specific problem is an appropriate approach to design and development activities.

5.2.2.5. Promising Interaction Techniques and Technology (R.11)

Description.

269. The bottleneck in the usefulness of interactive systems lies in the user-computer dialogue, the communication of requests and results between the user and computer. Thus interface design is critical. Making the interface intelligent, i.e., designing it to flexibly adapt to the user, the task, may be one way to cope with the communication problem. Other possibilities for enhancing dialogue are faster, more natural, and more convenient means for information exchange.

Content.

Intelligent Interfaces

270. The term intelligent interface is normally used to refer to user interfaces that respond flexibly, and adaptive to events in some purposeful way. The notion behind adaptive interfaces is that they adjust to the characteristics of :

a) an individual user
b) a specific task
c) a specific situation or environment.

271. It is widely accepted that such an adaptation requires the interface to maintain a user model embedded in the system (Murray, 1987). One of the fundamental steps in designing an adaptive interface is determining what aspects of the system will change in response to events or changing conditions. The following are some of the ways in which the system may adapt (Malinowski, Kuhme, Dietrich & Schneider-Hufschmidt, 1992; Rouse, 1988):

a) task allocation or partitioning - the system itself performs the complete task or parts of it;
b) interface transformation - the system adapts to make the task easier by changing the communication style and the content and form of the presented information;
c) functionality - the system adapts the functions available to each user;
d) user - the system can help the user to adapt by determining apparent problem areas and providing intelligent tutoring for them.

272. Another issue in developing an adaptive interface is determining the conditions that should cause the system to adapt. For example, the system might adapt to any of the following characteristics of the user, the task, or the environment:

a) user experience with the system,
b) user experience with the task itself,
c) user aptitudes (e.g., visual acuity or spatial reasoning ability),
d) user preferences (e.g., for special dialogue techniques),
e) task complexity,
f) task frequency,
g) probable workload.

273. The use of each adaptation criterion requires at least one source of data or some kind of model embedded in the system (Meyer, Yakemovic & Harris, 1993). Norcio & Stanley (1989) provide a literature survey and perspective on adaptive human-computer interfaces, presenting a survey of recent research as well as a discussion of factors that require consideration.

Multimodal and Multimedial Dialogue Design

274. The development of multimedia interfaces and dialogues requires a range of integrated software tools which support both the production of the information itself as well as the representation and communication of information. This component that controls and synchronises all media and varies the presentation mechanisms according to the specifications of a user- and task-model is critical.

275. A layered architecture may be an appropriate approach offering great flexibility for the application of multimedia technology. The lowest level provides information by means of different media and different qualities representing the knowledge which should be communicated to the user.

The next level provides knowledge about the interaction with the user permitting different interface presentations for the same interaction. At the third level deals with the integration and the course of interaction. On the meta level is the knowledge about the user. This architecture permits a determination of the dialogue structure, the dialogue objects, and the representation of the relevant information depending on the knowledge about the user and the task (Koller, 1992).

276. Jacob et al. (1993) give a thorough overview of the state of the art and current research opportunities in human-computer interaction styles and devices. They describe 17 different new input device issues and 15 output topics together with their proposed feasibility and utility.

Rationale.

277. Because of the rapidly evolving advances with reference to research and development in software and hardware technology a short outlook on most actual and future capabilities and feasibilities might be useful for interface designers to recognise and become familiar with possible trends.

5.3. EVALUATE

5.3. EVALUATE
- 5.3.1. Evaluation
- 5.3.2. Evaluation Activities during ANALYSE
  - 5.3.2.1. Verify Behavioural Model
  - 5.3.2.2. Verify Cognitive Model
  - 5.3.2.3. Verify Cognitive Requirements
  - 5.3.2.4. Verify Performance Measures
- 5.3.3. Evaluation Activities during DESIGN
  - 5.3.3.1. Verify Solution Concept
  - 5.3.3.2. Assess Adequacy of Design
- 5.3.4. Summative Evaluation Activities
  - 5.3.4.1. Assess Task and Organisational Performance
  - 5.3.4.2. Evaluate Decision Aid Usability
  - 5.3.4.3. Evaluate Technical Performance

5.3.1. Evaluation

Description.

278. The purpose of the evaluation activity is to refine requirements and to determine whether a decision aid is useful, usable, and used. The evaluation activity should occur parallel to design and, early on, should include a planning stage that considers all the purposes that evaluation is to serve. The overall planning of evaluations requires considerations of issues outlined in the Methods below. Evaluation activities are described according to the intended purpose and use of information.

Products.

279. The product of the evaluation will be information to make programmatic (strategic management) and design decisions. Information obtained from the evaluation can include the identification of problems, causes, and fixes and tracking of whether decision aid design objectives have been or are being met.

Methods.

280. The term "evaluation", as it relates to decision aids, does not refer to a single activity or type of activity. Instead, there are several purposes for engaging in evaluation activities, each with its own methods and procedures. Some of these evaluation purposes include:

a) Identification and verification of decision aid requirements;
b) Assessment of the technical aspects of the decision aid (including validity of content);
c) Evaluation of the performance of a decision aid;
d) Assessment of the usability of a decision aid (including acceptance by users);
e) Gathering data regarding the continued development and fielding of a decision aid.

281. In a general sense, evaluation purposes and activities can be categorised as a phase of development of a decision aid. That is, there are specific evaluation activities associated with the ANALYSE and DESIGN phase, and the Summative Evaluation phase of decision aid development. Evaluation methods associated with each phase are described in more detail in Sections 5.3.2, 5.3.3, and 5.3.4, respectively.

282. Another important aspect of evaluation involves evaluation planning. There are two aspects to evaluation planning. One aspect involves the technical issues of how to conduct the evaluation, and what will be measured. The second aspect involves administrative issues including scheduling, required resources, logistics, and the like.

283. In general, subactivities in evaluation planning include:

a) Decomposition of evaluation subgoals (according to the phase of development);
b) Identification of evaluation requirements and constraints;
c) Establishment and refinement of evaluation criteria (with emphasis on cognitive aspects of the task);
d) Selection and/or development of the evaluation methods (which are associated with the purpose of the evaluation activities). These may include:
   - Multiattribute utility theory
   - Cost-benefit analysis
   - Simulation and modelling
   - Experimental methods
   - Quasi-experimental methods
   - Survey data collection
   - Case studies
   - Task expert interviews
e) Preparation of evaluation materials and procedures;
f) Coordination with other activities;
g) Scheduling of evaluation activities.

284. As noted, the specifics associated with each of these subactivities will depend on the purpose for which the evaluation is being conducted.

Relationship to other activities.

285. Evaluation should be considered as an integral part of all phases of decision aid design. As such it is related in some fashion to all COADE activities. For example, during the Analysis phase, evaluation activities are required to verify the outcome of various task and cognitive analyses. During design, evaluation is necessary to verify and assess the design concept and implementation. During the Evaluation phase, assessment of the efficacy and usability of the aid is necessary, along with an indication of the impact of the aid on the task and organisation. It should also be noted that most evaluation activities recommended here should be considered iterative. That is, evaluation should be conducted in conjunction with other activities on a continual basis.

Rationale.

286. Evaluation is crucial to the success of a decision aid. If conducted correctly, an evaluation can be proactive. That is, it can help to identify and solve analysis and design problem before development has proceeded (and resources invested). Evaluation can also be diagnostic. In this sense, an evaluation can provide detailed information regarding the reasons why a decision aid is successful or unsuccessful. For example, if a decision aid fails because it is difficult to use, this suggests effort to improve usability. On the other hand, if the decision aid suffers from basic design flaws in functionality, investing in improving its usability will be misdirected and futile.

5.3.2. Evaluation Activities during ANALYSE

Description.

287. A number of evaluation activities should be conducted as part of the analysis phase of decision aid development. These relate to verification of crucial analyses, including accuracy and completeness of the behavioural model, cognitive model, cognitive requirements, and performance measures. Early assessment of these factors can avoid costly errors later in the decision aid development cycle.

Product.

288. The product of evaluation activities in the analysis phase include:

a) verified behavioural model
b) verified cognitive model
c) verified cognitive requirements
d) verified performance measures.

These products enable a decision aid designer or developer to move to the Design phase with increased confidence that the performance problem has been diagnosed adequately, and measures of performance specified sufficiently.

Methods.

289. The methods associated with analysis phase evaluation activities vary as a function of the particular purpose of the evaluation. These are delineated in more detail in the following sections.

Relationship to other activities.

290. Evaluation activities and associated data collected during the analysis phase have significant implications for other phases of decision aid development. With respect to the design phase, evaluation activities conducted in Analysis are crucial for articulating and understanding fully the problems in operational performance, and moreover, for determining which of them are amenable to decision aiding. Early evaluation activities can help a developer determine whether it is even feasible to build a decision aid, and if so, exactly how it should be designed. Obviously, having this type of information early in the process can help ensure that costly mistakes will be avoided, and can increase the probability that the aid will address and improve real performance problems.

Rationale.

291. If design activities are allowed to commence without assessment of the adequacy of problem specification, it is possible that the wrong performance and wrong cognitive problem will be addressed by a decision aid. Evaluation early in analysis can help to avoid design flaws caused by inadequate, incorrect or incomplete Analysis phase output.

5.3.2.1. Verify Behavioural Model

Description.

292. The output of several analyses recommended in the Analysis phase is a description in behavioural terms of the goals of a task, and how it is to be completed. This description is crucial because it specifies the analyst's perception of the manner in which a task is accomplished, and drives the specification of performance measures for the task. Moreover, it can guide the design of the decision aid by establishing task goals, and delineating procedures -- in this sense it encompasses the analyst's (implicit or explicit) assumptions about how the task should accomplished, and more importantly, can drive the decision aid design process. Therefore, it is important to verify the behavioural model of the task prior to the design phase.

Product.

293. The product of this type of evaluation is a verified or validated model of task behaviour. Unfortunately, methods to accomplish this are not always well developed; furthermore, those methods that do exist may be too costly (in terms of time or funding) to conduct. In such cases, it may be that an informal assessment of the behavioural model is all that can be completed. An informal assessment may entail consulting task experts to obtain their opinion regarding whether the behavioural model faithfully represents task demands.

Methods.

294. A variety of methods may be useful to verify the behavioural model of the task. At the simplest level, an analyst can present the proposed behavioural model to task experts to garner their opinion. More formally, if the task is modelled (e.g., using Petri nets, SAINT, or similar modelling techniques), computer simulations using such models can help to determine if the task has been accurately represented. Several techniques exist to aid in this process. For example, computer simulation algorithms can be useful as a means to predict performance as a function of prespecified parameters. In this way, the evaluator can test hypotheses or predictions about performance to determine whether the behavioural model is a faithful representation of actual behaviour.

295. Another potential method for verifying the behavioural model is to test predictions from that model using experimental or quasi-experimental techniques (see Adelman, 1991 for a description of such techniques). For example, an analyst may expect certain changes in task outcome as a function of particular task procedures; these can then be tested empirically.

Relationship to other activities.

296. Obviously, this activity is related to the Analysis phase, when the behavioural model is being specified via various analyses. With respect to timing, it should be noted that the activities specified here should be conducted in conjunction with Analysis phase activities. That is, verification of the behavioural model should be an iterative process that continues until the analyst has sufficient confidence in the accuracy of the behavioural model.

Rationale.

297. If the behavioural model is misspecified, it will suggest inappropriate performance measures and design solutions. Moreover, the behavioural model developed for a task is actually a precursor for the cognitive model because it describes in a global sense what should take place in task accomplishment. For this reason, early verification of the behavioural model is essential.

5.3.2.2. Verify Cognitive Model

Description.

298. As with verification of the behavioural model, the cognitive model must be verified to ensure that appropriate design decision are made. Here again, the analyst devises his/her conception of how the task is completed -- this time in cognitive terms. This conception (which actually represents a particular view regarding cognition and its impact on how the task is performed) must be tested before moving to the design phase. Unfortunately, this may be easier said than done. Due to the nature of the subject (i.e., cognition is not observable) it is difficult to determine or test the particular cognitive processes that typify effective performance. However, this does not mean that efforts should not be made to verify proposed cognitive models; on the contrary, it suggests that particular care must be taken in doing so.

Product.

299. The product of this evaluation activity is a verified cognitive model of the task. At the very least, it ensures that the analyst has taken the time to determine whether the cognitive model can be considered valid. State-of-the-art techniques are not developed enough to garantee that a cognitive model is accurate. However, even informal assessment of the cognitive model may highlight potential problems.

Methods.

300. A number of methods are available that can aid in the assessment of a proposed cognitive model. At the simplest level, task experts can be consulted to determine whether the analyst's conception of the task is accurate. Structured interviews are useful for this purpose, where the analyst leads the task expert through a series of questions regarding the cognitive activities that are required for task accomplishment. It should be noted however, that expert decision makers may have difficulty in verbalising their cognitive processes. For this reason, methods to augment interviews with task experts should be sought.

301. Computer simulation is another method that may be useful in verifying a cognitive model. By specifying the variables and processes assumed to characterise the decision making process, it may be possible to compare computer-generated performance with actual human performance. If the model is specified accurately, the two techniques should yield comparable results. In the same vein, experimental and quasi-experimental techniques may be useful in verifying the cognitive model. Here again, it should be possible to generate hypotheses regarding cognitive performance based on the proposed model, and then test these empirically.

Relationship to other activities.

302. Before specifying the cognitive requirements of the task, the analyst must have a reasonable idea of the likely cognitive processes involved in decision making in the domain of interest. The evaluation activities outlined in this phase should be conducted continually during the Analysis phase to help ensure that design decisions are based on the best information possible.

Rationale.

303. As noted with regard to verification of the behavioural model, early specification of inaccuracies in the cognitive model can help to avoid future (costly) design errors.

5.3.2.3. Verify Cognitive Requirements

Description.

304. Once the behavioural and cognitive models are verified, the next set of evaluation activities involves verifying the accuracy and completeness of the cognitive requirements hypothesised to be associated with the task. Therefore, this phase of evaluation entails determining that the cognitive requirements generated for the task are accurate (i.e., the analyst has correctly identified crucial cognitive requirements for the task), and complete (i.e., important requirements are not being overlooked). This phase of evaluation is paramount to later success because it can identify problems in cognitive requirement specification before these affect the design process.

Product.

305. The product of evaluation at this point is a set of cognitive requirements that have undergone verification and validation, refinement (if necessary) and further verification until the analyst is confident that the correct cognitive requirements have been established. The criteria for such evaluation are highly task specific. That is, a set of task-related measures must be developed that allow developers to determine the nature of cognitive demands imposed by the task. For example, the time required to reach a decision may be negatively affected by workload. If time to reach a decision is an important performance outcome, then the developer may decide to design an aid that attempts to off-load some of the decision maker's tasks. Other potential categories of performance that have cognitive implications can be found in Section 5.2.1.2; they include:

a) performing at the right time
b) applying knowledge more quickly
c) broadening considerations when making a decision
d) adopting a new or better sequence of performance
e) adopting a more efficient process
f) ensuring a proper understanding
g) making application of knowledge and rules consistent
h) avoiding omissions
i) complying with standards
j) improving option or choice selection
k) improving design or search for a workable plan
l) predicting or forecasting future states accurately
m) mimicking expert performance
n) supporting comprehension by "direct" representation of spatial material
o) mitigating the impact of uncertainty
p) increasing accuracy
q) decreasing workload
r) extending memory capacity
s) enhancing access to information in memory
t) quickening storage and retrieval of information
u) improving the production of documents (e.g., reports)
v) increasing the "learning value" of the task through feedback
w) improving team coordination and communication

306. In addition, a series of cognitive limitations have been identified that may be useful in determining whether the cognitive requirements are sound. These can be found in Section 5.1.4.1.

Methods.

307. The methods associated with this type of evaluation are similar to those employed to verify the behavioural and cognitive models. In this case however, the goal of the evaluation is to verify that the proposed changes in cognitive performance (due to the decision aid) will actually have the desired impact on operational performance. This is a difficult prospect. Once again, however, it cannot be overstated that identifying problems prior to design can avoid disaster later in the development cycle. Therefore, effort invested in verifying the cognitive requirements during the analysis phase is well spent.

Relationship to other activities.

308. As with other aspects to evaluation, verification of cognitive requirements is best viewed as an iterative process that continues until the developer has confidence that the appropriate cognitive

problems (or opportunities) are being addressed.

Rationale.

309. Perhaps the most crucial evaluation activities are those associated with verification of cognitive requirements. This is due to the fact that the cognitive requirements assumed to underlie task performance feed directly into the design process. That is, the cognitive requirements describe what is to be aided; the design of the aid itself then follows directly from this information.

5.3.2.4. Verify Performance Measures

Description.
310. A final set of evaluation activities associated with analysis involves validation of performance measures specified for a task. Obviously, these are closely related to cognitive requirements. The difference here is that the emphasis is on task outcomes (i.e., measurable aspects of task performance), and standards associated with these outcomes. See Section 5.1.2.1 for a more detailed discussion of performance measure development.

Product.
311. The product of this phase is a verified list of performance standards or criteria that will be used to determine whether the decision aid has the desired impact on performance. More particularly, what is needed is a set of measures that describes how the decision maker will perform with the help of the decision aid.

Methods.
312. Many of the methods associated with verification of performance measures are incorporated in task and cognitive analysis procedures. In fact, it is not always clear where analysis activities end, and evaluation activities begin. What is important here is that some effort is made to ensure that accurate, complete measures of performance exist so that it will be clear whether or not the aid is successful in improving performance.

Relationship to other activities.
313. As noted, verification of performance measures is an integral part of analysis activities. In addition, the performance measures verified here will serve as criteria for formative evaluations (conducted during ANALYSE and DESIGN) and summative evaluations (in the final stage of EVALUATE).

Rationale.
314. It is impossible to determine the efficacy of a decision aid (or the potential efficacy of a proposed decision aid design) without accurate measures of performance. In addition, well developed performance measures can help a developer diagnose the causes of decision aid failure.

5.3.3. Evaluation Activities during DESIGN

Description.

315. During the design phase, evaluation activities should be ongoing to ensure that the design is properly conceived and implemented. Therefore, several evaluation activities are appropriate during the Design phase. These are described in more detail in the following sections.

Product.

316. The product of design phase evaluation are data to support that efficacy of the proposed design solution (given the cognitive requirements), and evidence that the design was implemented faithfully (i.e. as intended by the designer).

Methods.

317. A variety of methods may be useful in evaluating a decision aid during design. Selection of a particular method will depend on why the evaluation is being conducted.

Relationship to other activities.

318. As with other evaluation activities, Design phase evaluation should be considered iterative. Using input from the Analysis phase, design will proceed by conceptualising and verifying a design concept, and then assessing the adequacy of design given this concept. In addition, evaluation information gathered during the design phase can feedback to the Analysis phase. For example, if cognitive requirements were misspecified in the Analysis phase, this may become apparent in design when problems are uncovered. In a more general sense, data gathered regarding design adequacy and the link between cognitive requirements and design strategies can aid in future development efforts by providing crucial information to guide design.

319. With respect to the Evaluation phase, data collected during Design can help to eliminate rival hypotheses when evaluating the overall system. That is, if the decision aid was found to be sound in iterative testing, then system failure may be attributed to other causes (e.g., user resistance).

Rationale.

320. Advances in computing allow for rapid prototyping of design solutions. By taking advantage of this opportunity, a designer can collect crucial performance data while the decision aid is in development, rather than waiting to find out that the design is flawed (after time and effort have been invested).

5.3.3.1. Verify Solution Concept

Description.

321. Evaluating the design concept underlying a decision aid involves an assessment of whether the proposed design will actually meet the needs laid out in the Analysis phase. That is, assuming that the cognitive requirements have been adequately specified, the question then becomes, does the solution concept address these requirements?

Product.

322. The product of this phase of evaluation is a design concept that is clearly linked to the cognitive requirements. The nature of evidence to support this conclusion will vary. The point here is that some attempt must be made to assess the proposed design to ensure that it addresses important performance issues.

Methods.

323. Analytical assessment of costs, projections of effectiveness, and expert opinion may be the only means to ensure that a design solution is addressing the correct cognitive requirements. Beyond this, mock-ups or early prototypes of a decision aid may be tested. However, assessment of design adequacy should be made before investment in development (in terms of the time spent, resources spent, and effort expended) becomes too high.

Relationship to other activities.

324. Using the output from the Analysis phase (i.e., cognitive requirements), it is possible to state in specific terms what the decision aid should be able to accomplish. Given sufficient detail (and, of course accuracy) in this process, it should be possible to make judgements regarding soundness of the design concept.

Rationale.

325. As with other evaluation activities, the more data available to make design decisions, the better. In this case, potential pitfalls may be avoided if the design concept is verified prior to development.

5.3.3.2. Assess Adequacy of Design

Description.

326. Once the Design phase has commenced and development has begun, an iterative evaluation strategy is recommended so that the decision aid can be refined as it is developed.

Product.

327. The product of this phase of evaluation is actually feedback to be used to refine, update, continue or even abandon an aiding strategy.

Methods.

328. The primary method for evaluation in Design is afforded by a rapid prototyping approach to design. From an evaluation standpoint, such an approach can be useful because it can foster or support frequent, interim assessment of the aid. Briefly, taking advantage of rapid prototyping for evaluation would lead to a "build a little, test a little" strategy, whereby the design is implemented enough to allow testing, refined, implemented further, tested, refined, and so on. Modern computing technology allows for greater flexibility (and lower cost) in rapid prototyping, making it an even more attractive strategy for conducting formative (i.e., iterative) evaluations during design. Of course the true value of rapid prototyping will depend in large part on the performance measures used to assess prototype systems. Only when performance measures are accurate, complete, and appropriately specified will a viable assessment of system performance via a prototype be possible; therefore, half-hearted, informal attempts to gather reactions from users via a rapid prototyping approach is not likely to provide the type of evaluation data required to make an accurate or comprehensive assessment of the aid. For more details on rapid prototyping, see Section 5.2.2.2.

Relationship to other activities.

329. Obviously, the Analysis phase plays a crucial role in determining what should be aided, and how performance should be measured. Both types of information are crucial during design evaluations.

Rationale.

330. Given the opportunities afforded by computer technology, it is often less costly to implement an iterative approach to evaluation, rather than waiting until significant and irreversible design decisions have been made.

5.3.4. Summative Evaluation Activities

Description.
331. Once the bulk of decision aid design and development are completed, the decision aid can move into a more formal evaluation phase. In the broadest sense, it must be determined whether the decision aid actually accomplishes the goals it sets out to in terms of operational performance. This is known as summative evaluation, and is typically what is thought of when the generic term "evaluation" is used. In this phase, there are several categories of information that are necessary to make a full assessment of decision aid performance. These include assessment of the decision aid's impact on task and organisational factors, the usability of the system and the decision aid's technical performance. These are described in more detail in the following sections.

Product.
332. The product of the Evaluation phase is information that leads to conclusions regarding the decision aid's effectiveness on several grounds. If conducted properly, the data from an evaluation can be diagnostic -- that is, it can specify the causes of ineffective performance, and suggest appropriate solutions.

Methods.
333. Several methods exist as a means to determine the efficacy of a decision aid. As with other aspects of evaluation, the particular method chosen will be a function of the purpose of the evaluation.

Relationship to other activities.
334. Unfortunately, it may be too late to ameliorate problems in decision aid design that are not discovered until the Evaluation phase (which is why we advocate an iterative evaluation strategy). However, the results of an evaluation can provide crucial data for future endeavours, so that other developers can avoid similar pitfalls. For this reason, it is hoped that decision aid developers and evaluators will document their results, lessons learned, techniques and recommendations so that future analysis, design and development efforts can be enhanced.

Rationale.
335. There are many reasons why decision aid's should be evaluated systematically. One was just mentioned; namely, to document procedures for the benefit of future developers. Beyond this, development sponsors may insist that evidence for the decision aid's effectiveness be harnessed. Finally, evaluation can provide feedback to the analysis and design processes so that decision aid performance can be improved.

5.3.4.1. Assess Task and Organisational Performance

Description.

336. The goal of assessing task and organisational performance is to determine whether the decision aid is effective in improving task performance as anticipated. Equally important, but less salient, is the notion that evaluation must establish why a system is not effective. That is, evaluation must be diagnostic so as to suggest design improvements.

337. Also of interest at this point is assessment of any "side effects" of the decision aid. For example, the aid may have an impact on other tasks or roles, on team member functioning, on life-cycle and maintenance requirements and/or on training requirements. These must be considered when judging the efficacy of an aid. Finally, evaluation of the impact of the decision aid on the overall organisation must be conducted. It may be found, for example, that an aid has the expected impact on decision maker performance, but does not affect organisational effectiveness as anticipated. This can occur when the link between task outcomes and organisational effectiveness are not well specified in the Analysis phase.

Product.

338. The product of this phase of evaluation is a set of outcomes that indicate 1) whether a decision aid is successful in meeting its stated goals, 2) whether organisational performance is improved as a function of the aid, and 3) whether the decision aid has unanticipated effects (either positive or negative) on other aspects of organisational functioning.

Methods.

339. At this phase of development, evaluation should be as empirical as possible. That is, controlled experiments or quasi-experiments should be employed when ever possible to draw conclusions regarding decision aid effectiveness. Obviously, "real world" constraints my preclude conducting controlled experiments. However, a number of techniques have been developed that approximate experimental control in field settings, allowing for acceptable hypothesis testing (see Cook & Campbell, 1979) In addition, Adelman (Adelman, 1991; Adelman, 1992) adds that case studies may be an effective means to evaluate decision aid performance.

Relationship to other activities.

340. The ultimate test of analysis and design procedures occurs in the Evaluation phase. In a sense, this set of evaluation activities is the culmination of other phases. It employs the performance measures established and verified in the Analysis phase, and tests the final design solution. In addition, data gathered here can feedback to improve procedures employed in these earlier phases.

Rationale.

341. Providing evidence of a decision aid's effectiveness is crucial to convince sponsors and users that the system is worth fielding and using. Moreover, as noted, a diagnostic evaluation can provide data so that the system can be improved.

5.3.4.2. Evaluate Decision Aid Usability

Description.

342. This type of evaluation deals more specifically with the usability and user acceptance of the system. Of primary concern here is whether users of the decision aid can employ the system in a reasonable manner. Issues associated with human-computer interaction are of interest here, along with data regarding the ease with which users can be trained to use the system. In addition, it is essential to assess whether the user population will accept the aid (i.e., their reactions to it), which will have an impact on their motivation and propensity to use it.

Product.

343. The product of this phase of evaluation is data to support the usability of the system, and to indicate that user motivation is sufficiently high to support use of the system. Specific criteria associated with usability and user acceptance include: confidence, ease of use, acceptability, extent of use, ease of training, and documentation (Riedel, unpublished) In addition, the quality of the user interface and match to the user (i.e., preference, level of expertise, background) must be assessed.

Methods.

344. Questionnaires, interviews and observation are primary mechanisms for assessing usability. For example, the developer may provide users with an opportunity to use the system, and then assess frequency of use as an indicator of user reaction. The developer may also observe user performance with the aid (either live or on tape) in order to get a detailed description of how easily the user interacts with the system. Along the same line, (unobtrusive) methods to assess the user's ability to work with the system (e.g., by recording incorrect data entries or button presses) may provide useful data regarding usability. Alternatively, user reaction (and associated variables) may be assessed more directly through interviews or questionnaires.

Relationship to other activities.

345. Data regarding usability and user acceptance provide pieces to the overall evaluation "puzzle". That is they are sources of information that allow developers and sponsors to make decisions about overall decision aid effectiveness. It should be noted that iterative evaluation in the design phase should greatly enhance the probability that usability issues are not a problem.

Rationale.

346. In evaluating a decision aid it is crucial to determine whether the user interface has been designed adequately. Without such information, it is impossible to rule usability or user acceptance out as possible causes of system failure. Moreover, even a well designed aid can fail when fielded if users refuse to use it.

5.3.4.3. Evaluate Technical Performance

Description.

347. This type of evaluation is concerned with the system's technical soundness. It includes such issues as: is the knowledge base complete, accurate and consistent; is coding to standard; is the system flexible; is the program portable; is computer usage appropriate; and is the system expandable (see Riedel, (unpublished report) for more detail). These data are important in determining overall decision aid effectiveness.

Product.

348. The product of this phase of evaluation is information that indicates that the decision aid's design has been appropriately implemented. A number of specific criteria exist in this regard; see Riedel, (unpublished report) for a more complete listing.

Methods.

349. One means of assessing the technical performance of a decision aid is to rely on expert opinion. That is, since clear standards do not exist for some of these factors, it may be that the best source of evaluation information rests with other developers and designers. Other, more objective evidence in support of the aid's technical performance can also be gathered; for example, data base design, data accuracy, algorithm accuracy and appropriateness, program portability, program flexibility, program efficiency, and the like can all be evaluated. In addition, any industry or professional standards that apply to a particular aspect of the decision aid's design (e.g., use of colour, character size, symbology, etc.) can be employed to assess technical performance.

Relationship to other activities.

350. Evaluation activities associated with technical performance specification can have an impact on the design phase by suggesting alternative implementation strategies to accomplish similar design goals.

Rationale.

351. Here again, data collected regarding technical performance can be seen as one category of information needed to assess fully a decision aid's performance. In addition, the design and development process of future efforts can be improved by studying in detail the impact of technical factors of overall system performance.

CHAPTER 6. COADE BACKGROUND

6.1. Models of Decision Making
6.2. A General Model of C2 Decision Making
6.3. Cognitive Task Analysis
6.4. Cognitive Performance Analysis
6.5. Performance Aiding Strategies and Guidelines
6.6. Human-Computer Interface Design
6.7. Evaluation

352. The field of cognitively-centred system development is quite immature and the concepts it draws in are wide-ranging and somewhat poorly defined. During development of the COADE framework (as presented in Chapter 5) it became clear that additional background material and a more extensive discussion on the activities and concepts underlying the framework might assist the reader in the application of COADE. This material is presented separately, in this chapter, partly to keep the framework itself from becoming too large and unmanageable. The chapter provides elaborations on COADE activities and concepts on analysis, design and evaluation as well as additional reference information.

353. RSG-19's view on decision making in C2 has evolved during the work of the group into one that considers decision making quite broadly, from the perspective of cognitive theory in general. Many of the activities in COADE require this cognitive perspective and its accompanying set of concepts and vocabulary. Two versions of this are provided in this report. Appendix A is a broad review of the concepts in cognitive psychology that are relevant to decision making, intended for the reader who is interested in some of the background theory. This treatment covers memory and knowledge representation, as well as the processes of thinking, and their control. Schema-based reasoning is covered in some depth, since it was this view of decision making that the group eventually adopted as its own perspective. The pertinent cognitive and schema concepts are highlighted in Section 6.2 of this chapter and developed into a generic model of decision making in C2. This model can be used as a backdrop for the subsequent amplifications in this chapter on cognitive task analysis, performance analysis and limitations. The generic model draws from current theories in the field of decision making as well as from theories in cognition. By way of further background, a brief review of decision making theories is provided as introduction to this chapter, in Section 6.1. This section contrasts the older "option-selection" theories with the more recent "naturalistic" ones. The latter attempt to take fuller account of the task and environmental factors that people, including military decision makers, encounter. It is intended that the generic model provide a view of decision making that is useful for analysis of C2 tasks. Section 6.2 therefore describes decision making in C2 terms and attempts to take account of some the factors identified by the naturalistic theories that may influence the process, especially familiarity.

354. The sections on the analysis of the cognitive processes and knowledge involved in a task (6.3) and the subsequent analysis of the quality of the cognitive performance (6.4) give further amplification on the processes in COADE that eventually result in the specification of cognitive requirements. The step from detailed problem specification (cognitive requirements) to solution is a large one. There does not seem to be a direct link between a requirement and potential solutions. Therefore, the section on performance aiding strategies (6.5) presents a taxonomy of support concepts that can help the analyst find adequate solutions. If the solution involves computer-based support, the section on human-computer interface design (6.6) can provides further guidance on the issues involved at that level. The section on evaluation (6.7) gives further rationale for the control of the quality of the intermediate results of analysis and design.

6.1. MODELS OF DECISION MAKING

6.1.1. Classical Models
6.1.2. Naturalistic Models

355. This section provides a brief review of theories and models of decision making, as background to the generic model of C2 decision making that is presented in the next section. The models can be roughly divided into the "classical" ones that regard decision making as a single-event choice between pre-generated options, and the recent "naturalistic" models that seem to better accommodate the loosely-defined task and dynamic decision environment of C2. The emphasis in coverage will be on the latter.

6.1.1. Classical Models

356. Classical models view decision making as a task of making a choice between several options or courses of actions. Typically the set of options is already available to the decision maker. The fundamental assumption is that people are "rational" and will not intentionally select an option that they know is inferior to another. Much of the work in classical model development has been on the specification of strategies that people use in different situations for considering the advantages and disadvantages of the options.

357. An early model proposed that people choose the option that maximises the "expected value" of the set of options. This basic model assumes that each option in the set has identifiable potential outcomes associated with it; these can be thought of as advantages and disadvantages for the option; each has a certain payoff or value (positive or negative) for the decision maker. The outcomes have known probability of occurrence if that option is selected. The expected value of the option can be calculated by summing the value of the outcomes associated with the option weighted by probability of occurrence. The option with the highest expected value is the one that should be selected.

358. An important theoretical development was the replacement of "payoff", which is an objective measure, with "utility", or subjective value. Similarly the objectively-measured probability of occurrence was replaced by subjective probability, i.e., degree of belief. These modifications to the model better accommodated individual differences between decision makers. The resulting strategy or rule was called "maximisation of subjectively expected utility".

359. A large number of such logically based option selection strategies have been observed in laboratory studies. Some examples are:

a) Dominance strategy, where the rule is to choose the option whose utility is at least as attractive as that of every other option for all attributes of interest and better than every other option on at least one attribute (Lee, 1971).
b) Elimination by aspects, where options are judged on one attribute at a time; any option that fails to meet some pre-set criteria on that attribute is eliminated from the set of possibilities; the process continues iteratively until a single option remains (Tversky, 1972).

360. Zsambok, Beach and Klein (1992) have categorised 15 of the principle option-selection strategies identified in the literature according to the conditions under which they are relevant:

a) the goal of the option selection (e.g., selection of the best option, selection of an

acceptable option, the screening out of unacceptable options);
b) characteristics of the options' attributes and outcomes (desirability, reliability, completeness, measurement properties);
c) application requirements (single or dual strategy, type of evaluation -- relative, absolute, compensatory, noncompensatory);
d) environmental circumstances (availability of a decision aid, degree of problem structure, time available to make the decision).

361. Pounds and Fallesen (1994) have classified 66 problem solving strategies into three classes, each with three subordinate categories:

a) Managing information:
   considering hypotheses, beliefs, uncertainty.
   combining information.
   managing amount of information.
b) Controlling progress:
   ordering by hierarchical structure.
   sequencing.
   ordering by merit or payoff.
c) Making choices:
   managing the number of options.
   using compensatory choice.
   using noncompensatory choice.

They concluded that many of the strategies identified from laboratory studies are not descriptive of strategies used in everyday problem solving. "Everyday" strategies are less standard, ad hoc, being modified for the specific problem encountered.

362. Classical decision theories originated as a means of describing decision making performance in standard, objective ways. Many were based on Bayesian probability theory and provided a formal, mathematical rationalisation for the optimum choice from a set of options. Gradually some of the models started being used as prescriptive standards for evaluating a decision maker's choice of option, assuming that he or she was behaving rationally. The evaluation was made in terms of the internal coherence among beliefs, preference and actions. Thus the classical models evolved into benchmarks for evaluating performance (Cohen, 1993). Comparison of human performance to these "normative" models in various situations resulted in the identification of certain persistently-occurring patterns of errors — violations of the consistency constraints imposed by probability rules of the model. These errors were explained as "biases" in human decision making. Biases include errors in assessing probabilities (e.g., overconfidence in estimating the probabilities of simple events); errors in inference (e.g., disregarding or discounting evidence that conflicts with a prior hypothesis); errors in choice (e.g., the evaluation of later outcomes in a time sequence as if earlier uncertain outcomes were known for sure). A large set of such biases was identified, leading to the suggestion that humans behave "non-rationally", at least in certain circumstances. The biases were an attempt to give an explanation of the reasoning (or failure in reasoning) process in decision making, but as Cohen (1993) states, the account was in terms of "a diverse set of unrelated cognitive mechanisms".

363. Although the problems used to study biases in decision making were drawn from real life (e.g., buying a house), in contrast to the gambling problems used in earlier research, the decision maker's task was still presented as pre-structured (in terms of the option set) and pre-quantified (in terms of probabilities and payoffs); and the task usually involved a single response to a static situation.

364. The evidence is now mounting that people making decisions in everyday problems do not typically use the strategies suggested by the normative models. A body of research on "everyday reasoning" has developed in response to the criticism that findings from the laboratory may not be generalisable or applicable to real-world settings. Everyday problems have been characterised as those problems having an ill-defined structure and multiple workable alternatives (Meachan & Emont, 1989).

6.1.2. Naturalistic Models

365. Models of everyday reasoning have been developed to address reasoning which is based on inferences that are potentially falsifiable. Rather than following a well-controlled linear syllogistic reasoning process, this type of reasoning takes advantage of the problem solver's prior experiences. Strategies appropriate for the current situation may have already been developed a so can be reused, making unnecessary a deliberate analysis of the current situation as a new situation (Perkins, Jay & Tishman, 1993). Information from past experiences is incorporated into the reasoning process; inferences are compared to subsequent information and may be overturned (Holyoak & Spellman, 1993).

366. Research of everyday reasoning has recognised that the problem's structure, options, and goals are relative to the problem solver's experiences. The different view of decision making that is emerging emphasises the features of the task and the decision maker's knowledge and experience relevant to the task. This view, embodied in "naturalistic" theories of decision making, is an attempt to move the decision task back into a meaningful real-world context, where the decision is not an end itself, but a means to achieve a broader goal.

367. Orasanu and Connolly (1993) have identified eight important factors that characterise naturalistic decision making:

a) The decision problem is ill-structured; there may be more than one way to solve it;
b) The decision environment is uncertain and dynamic; information may be incomplete or ambiguous or misleading (as is typical in C2); the environment may change rapidly;
c) There are shifting, ill-defined or competing goals;
d) The situation involves a series of events consisting of action and feedback, rather than one single decision point;
e) Decisions are made under time pressure; this may induce high levels of personal stress, and change the reasoning strategies used;
f) There are high stakes associated with the outcome of the situation;
g) The decision may involve several individuals acting in a team, possibly geographically distributed;
h) There may be an organisational setting that provides or imposes goals, rules, guidelines and values that influence the decision making process.

368. Many of these characteristics are typical of C2 situations, and for this reason, the naturalistic theories seem particularly relevant to COADE. These theories, which are well summarised in Klein, Orasanu, Calderwood, and Zsambok (1993) are briefly described.

369. Klein's model of "Recognition-Primed Decision Making" (Klein, 1993) is representative of the major differences between classical and naturalistic decision models. It is a descriptive model that focuses on the behaviour of experienced decision makers who may not have much time to make their decisions. The model assumes that decision makers who are expert in a domain have available mentally stored "prototypes" of previously-experienced situations. These prototypes (or schemas) contain the distinguishing characteristics or cues for the situation, the goal(s) that are associated with the situation, and a set of actions (course of action) that has been successfully used in the past to satisfy the goal. The prototypes also contain expectancies about how the situation should evolve if the course of action is implemented. The model proposes that decision making consists primarily of assessing the real-world situation and finding the prototype that best matches it. The triggering of a prototype is a pattern-matching process that can happen almost automatically, based on the characteristics of the real-world situation. If there is little time in the decision situation, and the match between the prototype and the current situation seems good (i.e., the situation is "familiar"), then the course of action associated with the prototype may be implemented directly. If time is available, and especially if there is some doubt about the suitability or completeness of the course of action, the decision maker will mentally simulate or "play out" the steps to be taken. This process identifies problems and permits modification of the course of action. The result may be acceptance of the course of action, or rejection of the course of action and further assessment of the situation to determine an alternative option.

370. David Noble's model of decision making through Situation Assessment (Noble, 1993) is similar to Klein's, in the sense that it emphasises the importance of development of a mental representation of the real-world situation. The representation is developed by observing concrete characteristics of the situation, and coupling these with the contextual background knowledge surrounding the situation ("implied" characteristics) and general knowledge about such situations retrieved from memory. The representation so-formed is used to generated expectations about the way that the situation will develop, and these can be tested through further observation or information gathering. If the expectations are confirmed, then the decision maker may adopt the course of action that has been successful in similar situations in the past. Again the important feature of this model is the stress on the decision maker's iterative assessment of the situation, rather than simple option selection.

371. Pennington and Hastie (1993) have offered a somewhat different view of situation assessment during decision making, but one that is in line with Klein's and Noble's. According to their model, which was developed through observations of jurors determining the verdict in a trial, decision makers develop a story or "explanation" of the events in a situation, similar to a schema. The elements of the schema are connected by causal relationships that explain why the events occurred (for example, the participants had certain goals that caused then to undertake actions) and temporal relationships that order the events. Although the evidence describing the situation may be incomplete, a more complete and coherent story is developed by the decision maker by inferring a portion of the events to fill in the gaps. Pennington and Hastie argue that this type of explanation-based reasoning is prevalent when people must process a large amounts of situation information that only partially describes the situation; and when the information about the events in a situation is not presented in chronological order.

372. In contrast to the preceding models, Montgomery's (1993) addresses the question of how a selection is made between several options. He suggests that decision makers do this by searching for "dominance structure". Rather than a completely weighing all options in a parallel manner, Montgomery proposes that people choose one promising alternative (perhaps based on a particularly important attribute, like cost) and then do a quick check to see if that alternative dominates the others. Domination means that the alternative is at least as attractive as the others on all attributes, and is superior on at least one attribute. If not, the decision maker tries to adjust its relationship by de-emphasising the importance of inferior attributes; enhancing the value of the superior attribute by weighing it more heavily; trading off the advantages and disadvantages of attributes; or integrating several attributes into one. If none of these strategies works, the decision maker chooses another alternative. This view of decision making as having an initial quick selection between alternatives is similar to Klein's.

373. Image theory has been developed by Beach and his colleagues (1990) from observations of a wide variety of real-life decision making tasks. It is broader in its scope that the previous models, accounting for the principles and values of the decision maker as well as past experience in similar decision situations. The theory assumes cognitive structures (schemata) called "images" that organise the decision maker's goals and knowledge in increasing level of specificity. The most general, the value image, contains the fundamental, relatively-permanent principles (standards, ideals, morals, beliefs and ethics) that guide overall behaviour. On the next level of specificity is the trajectory image, an agenda of goals that the decision maker has decided to pursue, together with related time-lines for achieving them; the time lines may have associated marker events that are indicators of intermediate progress. Finally, the strategic image holds the plans and tactics the decision maker has for accomplishing the goals and the decision maker's forecasts of the effects of implementing the plans in terms of the chances of successfully attaining the goals. The theory speaks in terms of the "framing" of decisions, rather than the "making" of decisions. Framing involves the identification of a goal and the associated recall of plans that have been formulated in previous situations for achieving this or a similar goal. This brings into mental focus a manageable subset of the knowledge in the images (the "frame"), that provide inferences about information that is not immediately apparent; decisions serve to change the frame appropriately as the context evolves; the frame is the baseline against which all change is evaluated.

374. There are two types of decisions in image theory. Adoption decisions concern the addition of goals and plans to the decision maker's current agenda, and they are made first and foremost on the

basis of whether the plan is "compatible" with the framed images. This process screens out unacceptable goals and plans. If more than one candidate passes the screening, the most profitable one is then selected (through use of one of variety of strategies). Progress decisions determine whether an implemented plan is actually achieving the objectives; this type of decision making is also used to "play through" a plan proposed for implementation, to determine whether it is compatible with the constituents of the trajectory image.

375. Rasmussen's model of decision making (Rasmussen, 1993) is derived from observations of humans controlling complex automated systems, and thus focuses on the assessment of dynamic situations. The theory proposes that behaviour and the accompanying decision making are controlled by qualitatively different mechanisms, depending on the degree of experience of the operator. Operators who are very experienced with a situation can act at a "skill-based" level: their responses to situation assessment are highly tuned and fluid and are invoked almost automatically. The responses are controlled in time and space at a perceptual-motor level. There is, in fact, no decision making per se at this level. When the situation is less familiar (and the operator is less experienced) behaviour comes under control of decisions that are consciously made. In this "rule-based" behaviour, the situation is recognised as being of a previously-experienced type, and the rules (or actions) that have been linked to it and stored in memory are executed. This is analogous to Klein's "recognition-primed" fast path. Novel situations preclude an automated or rule-based response and require explicit situation assessment. The "knowledge-based" behaviour at this level involves the construction of a mental model (based on causal and functional relationships) at different levels of abstraction, and the explicit creation of options. People attempt to keep the processing at the lowest cognitive level that assures trustworthy performance of the task.

376. The decision cycles model of Connolly and Wagner (Lipshitz, 1993) proposes that decision making consists of an interplay between situation assessment, evaluation of alternatives, and action. The decision maker holds two domains of information mentally: a cognitive map of the world which is adjusted by feedback from the consequences of actions taken; and a set of goals which are used to evaluate alternatives. The goals themselves may also be adjusted by feedback from actions. Connolly argues that decision making is a continuous process, and in his model acting and thinking are very much inter-related. He suggests that there are two kinds of decision processes, whose use depends on the situation. If the goals are well-defined, and there is a clear way of achieving them, the "tree-felling" strategy can be used: the decision is made in a single stroke. By contrast, the "hedge-clipping" strategy is used where the situation is uncertain, goals are ambiguous, and it requires less cognitive effort to proceed incrementally.

377. Hammond's theory of decision making (Hammond, Hamm, Grassia & Pearson, 1987) emphasises the difference between cognitive processes that he terms "intuitive" and those that are "analytical". A process is more intuitive and less analytical when it is executed under low control, low conscious awareness, and when there is a rapid rate of data processing and high confidence in the answer. In fact, he proposes that real-world decisions are made in a quasi-rational way, using a combination of intuitive and analytical methods. He argues that the degree of analytical vs. intuitive processing is governed by the characteristics of the decision task: for example, tasks that require the processing of large amounts of information in short periods of time induce intuitive methods, whereas tasks that provide quantitative information in a sequential fashion induce analysis. The theory proposes that analytical methods are not always best; that accuracy in judgement is highest when the characteristics of the decision task match the nature of the cognitive process.

378. In his study of military decision making, Lipshitz (1993) has conceptualised it as "argument-driven action" where the mode adopted for decision making corresponds to different kinds of arguments. Some problems can be framed as forward-looking choices between several alternatives; in this case the action alternative is selected on the basis of having the best consequences or outcome in the future. Other problems can be framed as situation assessment problems where experience provides a rule or guidance about the action to take. Finally, a third class of problems, reassessment, can be framed as objections to a certain course of action because of uncertainty.

379. Naturalistic decision models broaden the classical view of decision making, emphasising
a) the development of a representation of the situation that permits identification of plausible courses of action; and
b) the use of mental simulation for evaluation of options.

These models take into account that processes for decision making are context dependent; and that they depend on the experience of the decision maker, the values imposed by the organisation, the characteristics of the task and how information is presented. Decision making is regarded as a dynamic process involving, for example, a preliminary quick selection of a course of action followed by a more deliberate evaluation and expansion of it. These models do not treat decisions as single isolated events divorced from situation assessment, although they can accommodate the classical strategies such as elimination by aspects.

380. The naturalistic models seem to take account of many of the important characteristics of the C2 decision environment, and it is expected that they will form the basis for future theories of decision making in C2. In the next section (6.2) we will turn to a more specific description of the processes and structures of cognition that are involved in decision making in C2.

6.2. A GENERAL COGNITIVE MODEL OF C2 DECISION MAKING

6.2.1. Knowledge Representation & Memory
6.2.2. Monitor
6.2.3. Basic Processes
6.2.3.1. Understanding
6.2.3.2. Reasoning
6.2.4. Meta-cognition
6.2.5. Act
6.2.6. Learn
6.2.7. Situational Factors Influencing the C2 Decision Process
6.2.7.1. Familiarity - High
6.2.7.2. Familiarity - Moderate
6.2.7.3. Familiarity - Low
6.2.7.4. Effects of other factors

381. This section provides a general model of the decision process in C2 that can be used as a basis for cognitive task analysis and further development of a cognitive model specific to a decision situation. The model is based on the cognitive concepts and terminology identified in Appendix A, particularly that of schema-based problem solving. It also takes account of specific models of decision making discussed in Section 6.1. The model is generic, in the sense that it can be applied to any decision situation. It assumes certain cognitive components — a knowledge representation structure and set of basic decision processes — that occur to varying degrees in all decision situations. It is targeted towards modelling the cognition of a single decision-making entity and thus does not account for the interaction amongst members of a decision making team.

382. It is assumed that the decision maker is monitoring a military situation in the real world against which he anticipates implementing some as-yet-undetermined course of action (including, possibly "No action"). The timing of the actions (and the time horizon) is not limited; thus the need to act may be fairly immediate (e.g., within minutes, hours) or it may be further in the future (days, weeks).

383. The components of the decision system include the following structures and processes:

a) **Monitor** - processes for receiving information about the world and for monitoring the execution of plans;
b) **Act** - processes for performing action upon the world;
c) **Knowledge Representation (& Memory)** - structures for storing the history of actions and experiences; for buffering the reception of information and the effecting of actions;
d) **Basic Processes - Understand, Reason** - for interpreting and identifying information; for controlling actions; for modelling the environment and self; for devising plans;
e) **Meta-cognition** - processes for controlling the allocation of attention and cognitive resources;
f) **Learning** - processes for updating memory.

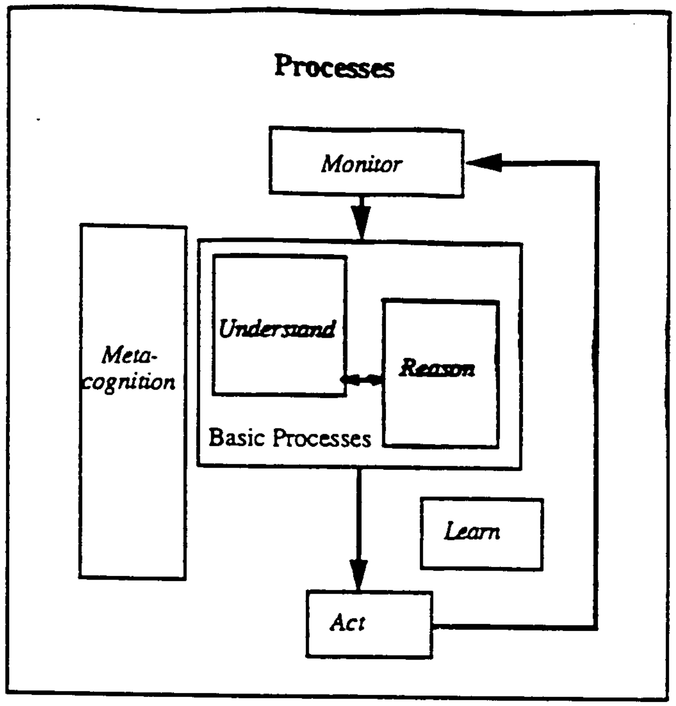

Figure 6.1. COADE Model of Cognitive Processes in C2 Decision Making

384. Figure 6.1 illustrates the processing part of the model. Briefly, the decision process acts upon knowledge structures (not shown in the figure) that are drawn from (or activated in) long-term memory; and knowledge structures that are created or modified based on the perception of information from the decision situation. These structures are organised in working memory. The detailed nature of these structures will be described following this overview. The decision process is a loop beginning with Monitoring of the external (decision) situation. Monitoring includes a degree of situation assessment to determine whether the external situation is developing satisfactorily (possibly based on expectations associated with the implementation of a previous course of action). If this is not the case, a more active analysis (Understanding) of the situation takes place, together with Reasoning about goals and constraints, and previous situations of a similar nature. This results in the development of one or more courses of action, which then may be actively evaluated against criteria or constraints. The central cognitive processes of Understanding and Reasoning take place largely in parallel. Ultimately a decision is made to implement one developed course of action (in Act) and the effects of the implementation are further Monitored, thus beginning the cycle again. Two additional categories of process take place alongside the main decision process: Meta-cognitive processes that are like an executive level of control, concerned with allocating resources; and Learning processes which update long-term memory, based on both the results of the decision process itself and the results of the course of action.

385. The degree to which these components and processes enter into decision making will depend on a number of factors, including the familiarity of the situation, the time available to make the decision, etc. The influence of these factors on the basic process will be discussed at the end of this section.

6.2.1. Knowledge Representation & Memory

386. At the start of the decision situation, the decision maker holds some knowledge in long-term memory that can be considered to be relevant to the external situation. The more often the situation has been experienced in the past, the more specific (and detailed) will be the knowledge that is relevant. If, on the other hand, the situation is novel, then the decision maker will be able to draw on only general and abstract information from long term memory to help in Understanding and in Reasoning about a course of action (COA). The knowledge available in memory coupled with the

information perceived from Monitoring the external situation, is what is manipulated and transformed by the various decision processes.

387. A useful distinction can be made between knowledge that pertains to facts, (concepts and relationships); and knowledge about how to do something. The former is called declarative knowledge and the latter procedural.

388. The fundamental component of declarative knowledge is the concept, a mental representation of a class or an individual. A concept may be available in memory as a name (a linguistic representation), an image (a pictorial or spatial representation) or even as a perceptually-grounded phenomenon such as a smell. The same concept may be stored in different ways. Associated with concepts are attributes, properties that characterise the concepts2. For example, a battleship has a physical length, a complement, certain kinds of weapons. Concepts include general categories and more specific categories and instances (e.g., a battleship is an instance of a surface vessel). Thus concepts can be considered to be organised in hierarchies based on specificity. This is portrayed in Figure 6.2. Furthermore members in a category may be represented by a prototype, a general concept having attributes that characterise the members of the category. However the prototype does not necessarily exist in the real world. Related to this idea is that of typicality, the degree to which a (real-world) instance is representative of a concept. It is also useful to think of concepts as being organised in networks, in which the concepts are linked by relationships. Relationships can be of many kinds, and reflect the formal and informal learning that takes place in a domain. The strength of a relationship is a measure of the degree of coupling or association between two concepts (e.g., battleships are part of certain naval formations).

389. Procedural knowledge is information on what actions to take when faced with different situations. This kind of knowledge can range from (consciously inaccessible) knowledge on the fine motor-control actions necessary for skilled physical performance (e.g., writing, firing a weapon) to high-level plans that specify a series of temporally-organised actions that can themselves be further detailed. Procedures can be overt actions on objects in the real-world (e.g., drive the tank across the stream; assault enemy position at location XYZ; etc.), or mental actions performed on information stored in memory, like deductions, calculations, and comparisons between two concepts. One kind of high-level mental procedure is the mental simulation of a proposed plan. Procedures use and transform declarative knowledge. It is convenient to think of procedures as subroutines (that have other procedures embedded within) that can be nested in hierarchies (according to abstraction) or at least linked in networks (Figure 6.2). A procedural hierarchy can be thought of as a plan with a top-level goal that can be achieved by executing actions at the next level of detail. However, those actions can also be considered to be goals, to be achieved by a further series of sub-actions, etc. So the actions at one level become the goals at another. Thus there is a tight linkage between a procedural hierarchy and the goal hierarchy. Typically, procedural knowledge has a temporal component since it usually dictates the order of execution of actions.

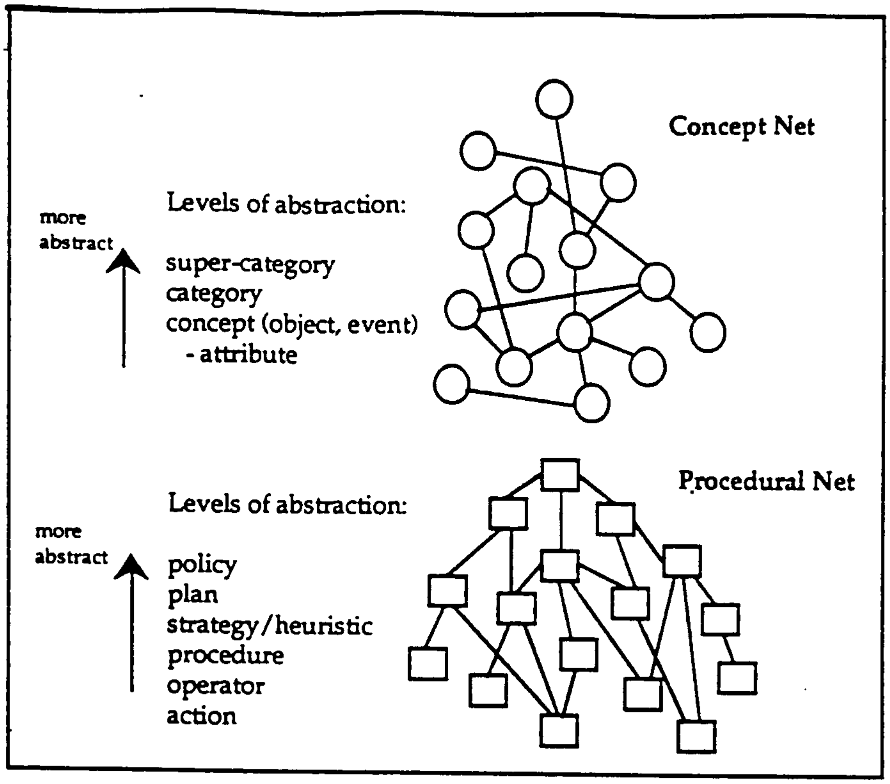

Figure 6.2. Levels of Abstraction in Concept and Procedural Nets

390. Different terms have been used to describe procedures at various levels of detail, but there is little consistency in their use. The following is a list ordered in what RSG.19 considers to be increasing level of abstraction, together with generally-accepted definitions where available:

- action - an activity to be undertaken in the world
- operator - a method for transforming a problem from one state to another
- rule - a method for representing knowledge about both concepts and procedures; rules can be used to place instances in categories and to make predictions about the way that category members will change over time in response to actions; rules are often represented by conditions and associated actions (as in production systems)
- procedure - a series of actions followed in a specified order
- strategy - a procedure for solving a problem; the term is used for both high and low-level procedures; a heuristic is a rule of thumb that guides search through a problem space to achieve the goal state, where the problem is solved; a simple efficient criterion used to narrow a set of choices; heuristics do not guarantee a solution
- plan - an anticipated sequence of actions that begins with goal adoption and ends with goal attainment (plans "try to" achieve some event or state); plans can be at different levels of detail
- policy - a general plan or guiding principle

391. A well-established view in cognitive psychology argues for the existence of a high-level knowledge structure called the schema. A schema is a cluster of information describing the characteristics of a situation, together with procedural knowledge about the typical procedure, sequence of events, actions or solutions associated with the situation. Thus the schema provides a structure for linking declarative knowledge based on experience (possibly in the form of a concept network) with a procedural network that gives the actions and sub-actions necessary to achieve the goal associated with the situation (Figure 6.3). Some models of decision making (notably the "naturalistic" ones) suggest that the schema stores information of a specific type that assists in recognising or describing problems and situations. Cues are important features or attributes of the

situation that serve to distinguish it from others; criteria are the part of the schema used for evaluating and weighing features; expectancies dictate how the situation will typically evolve over time. Schemata may be organised in networks, linked through their associated concept and procedural nets (Figure 6.4). As well, their knowledge is at different levels of abstraction, with general ones pointing to more detailed ones. The term schema has typically been used for structures that link a real-world situation and its characteristics to a suitable procedure. However, schemata need not necessarily be triggered by physical characteristics of an external situation; they could be invoked by cues that arise as a result of structuring a problem.

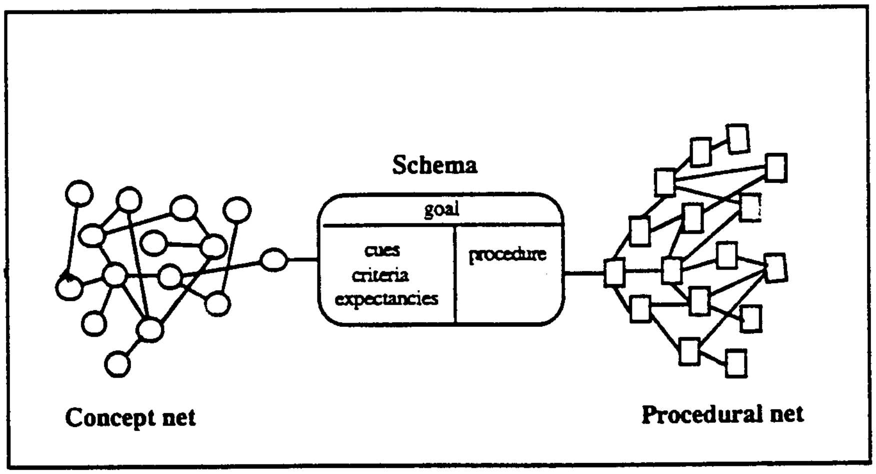

Figure 6.3. Schema Supported by Concept and Procedural Nets

392. Thus the knowledge in memory can be considered to consist of declarative knowledge such as concepts and categories of objects and events; procedural knowledge, including plans and actions that achieve goals; and schemata that link situations with typical actions. Situations and events external to the decision maker are constantly causing the triggering or activation of subnets of concepts, plans, actions and schemata in memory (Figure 6.4). The strongly-activated part of these networks that is the focus during problem-solving and decision making is called the mental model of the problem. The mental model is a dynamic working representation of the problem situation, derived from perceptual and verbal information and constructed when required. It can represent both physical relations and conceptual features and may be derived from (or include) schemata; it permits the mental simulation of the effect of possible courses of action.

393. Beyond the specific characterisation of the problem in terms of a mental model and the subgoals associated with parts of the model is a set of overriding goals, principles and values that guide the decision maker. Principles and values are the relatively permanent imperatives that prescribe what one ought and should do; they are also the general standards by which actions proposed as a response to a situation can be judged in general; they are mainly products of culture, and include standards, ideals, beliefs, and ethics.

6.2.2. Monitor

394. A scanning of the environment to determine either a) the need for intervention to alter the sequence of events to a desired direction; or b) the need to alter a previously-implemented COA because it is not having the desired effect (i.e., events are not proceeding in the desired direction). Monitoring includes processes like detection of feature, instance, and event, and the matching of situations features to expectancies.

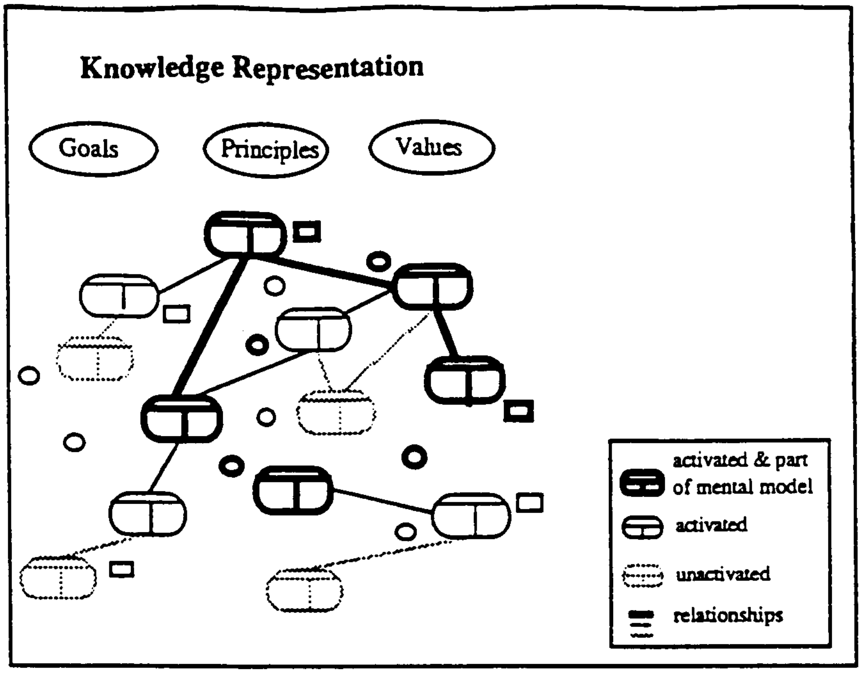

Figure 6.4. COADE Model of Knowledge Representation Used in Decision Making

6.2.3. Basic Processes

395. Current and traditional theories in decision making suggest that the process in C2 can be regarded as some combination of Understanding the decision situation in order to form a model of the current and expected state of the world and Reasoning using the model. The processes in Understanding and Reasoning are complementary.

6.2.3.1. Understanding

396. Understanding is the process by which an observed situation and its events and features are compared to previously-experienced situations in order to produce a mental model of the current situation. Part of the process of Understanding involves an intuitive process of pattern matching that results in the activation of a network of schemata. These schemata may be quite detailed and situation-specific if the situation is familiar. In this case, the "discovery" of the associated COA is automatic and almost without effort. In less familiar cases, Understanding may involve an implicit checking of the situation cues and expectancies provided by the model to confirm its validity. And it may involve a mental simulation of the linked COA to determine whether it is satisfactory for the current situation. When experience provides no situation-specific set of schemata, the process of Understanding requires the adaptation or modification of a more general schema by filling in the slots to make it match the parameters of the situation. This is called instantiation. Cases where the situation is entirely novel call for knowledge-based problem solving, in which the set of potentially applicable schemata is broaden to include analogous problems (also stored in schema form) and the procedures used for solving them.

397. The process of Understanding attempts to find or create the most detailed model of the situation that is valid. The term specialisation describes the rejection of a coarse-grained schema in favour of a more specific subordinate schema. The finer the grain of resolution of the model, the more completely the COA is specified, and the less effort that will be required to flesh out the COA. However, an invalid model at a fine grain of resolution may suggest the wrong COA.

6.2.3.2. Reasoning

398. The process of Understanding is driven mainly by episodic memory structures that reflect direct experience with decision situations that are the same as or similar to the current situation. The

processes of Reasoning are analytical and are centred around rules and procedures that have been taught or derived at an abstract symbolic level. Reasoning processes operate on the model that is developed from Understanding. They are logically-based, rigorous and complete, and can be time-consuming to execute. Typical kinds of reasoning include the operations of ordering concepts or events in a sequence, comparing and evaluating, testing, and choosing. The procedures characterised by "if ... then ... else" rules in production systems are another example of deductive operations.

6.2.4. Meta-cognition

399. Meta-cognitive processes are processes that sit at a different level than those described above. They are responsible for overall control of the decision process, and the allocation of the cognitive resources of the decision maker. So for example, meta-cognition determines the overall strategy used for making a decision, the setting and balancing of several overall goals, whether the mental model used for Reasoning is adequate, how the search for information for the model should proceed, how extensive and thorough Reasoning or Understanding should be, whether sufficient evaluation of the COA has been carried out. Meta-cognition also takes into account the limitations of the decision maker him or herself: the allocation of attention, the capacity of memory, the cognitive effort available, and the assessment of self performance.

6.2.5. Act

400. The implementation of the selected COA. This occurs under the control of the plan associated with the COA.

6.2.6. Learn

401. Learning is the updating of memory structures to reflect the results of the decision making both in terms of the process used and the success of the outcome. In some instances, Learning involves major restructuring of memory and the formulation of a new schema. In other instances it simply involves the recording of a new instance of an existing schema (accretion). More often, it involves the tuning of existing schemata, that is, the elaboration and refinement of the concepts or procedures linked in the schema. The cues and procedure in the schema associated with the decision making situation may be made more specific and detailed through a process of specialisation. Alternatively generalisation may occur. This is a process that increases the range of examples that are covered by some category or procedure. For example, the grouping of individual instances of concepts or events into classes (also referred to as classification) is a kind of structuring that then allows class properties to be used when reasoning about instances in the class. Procedures also may be integrated or compounded into a single procedure.

6.2.7. Situational Factors Influencing the C2 Decision Process

402. The precise nature of the decision process in C2 is governed by characteristics of the C2 situation itself. Three factors seem to be central. The first is familiarity, the degree to which the decision maker has experience with the decision situation, has seen it before, and has established a successful course of action (COA) to deal with it. Associated with familiarity is situation predictability, the degree of certainty that the situation is likely to follow a predictable sequence of events over the future. If the situation is unstable it is less likely to be predictable. The criticality of the decision is important, that is, the degree and extent of its potential impact, and the consequences of a wrong decision. Another central factor is the time available to the decision maker before action must be taken. This is partly a function of the situation, but may also be confounded by deliberate actions taken on the part of the decision maker to gain time. Note that these factors are not orthogonal.

403. Other situational factors also play a role. Potential side-effects of the decision may be an issue, to the extent that the decision maker can estimate them. The rate of feedback from actions taken in the environment will be a factor in determining whether the decision maker can act to get more information when time is short. Related to this is the number of opportunities for action available to

the decision maker. The extent to which the decision maker has worked out specific and well-defined goals and the degree to which these might be altered will play a role. The cognitive style of the decision maker may be an influence. The effect of group dynamics on decision making has not been explored, but it is certainly an additional factor.

404. The following is a description of how the decision process might be influenced by various combinations of the situational factors. The discussion is organised along the factor of familiarity, from high to low (reflecting an influence by recent theories of naturalistic decision making). The treatment assumes certain cognitive processes and structures (described in the previous sections 6.2.1-6.2.6), and the possibility of external aids to decision making, like a partial representation of the situation (e.g., a situation display) and/or means for representing and manipulating possible courses of action. The analysis has not been validated empirically, and so must be regarded as suggestive only.

6.2.7.1. Familiarity - High

405. Familiarity with the situation assumes that the decision maker has available in memory a fairly detailed schema to use as a prototype for action, one that at least partly matches the features of the situation. This prototype is activated automatically from long-term memory by an understanding process. Cues provided by the schema may be confirmed by deliberate comparison of the characteristics of the prototype with the characteristics of the actual situation to the extent that time is available. If the situation is anticipated to be reasonably predictable (and especially if time is short), the associated COA, which is well-defined, is implemented directly in Act. If there is not a good match between the cues suggested and the actual situation, or if the situation is anticipated to be unpredictable, the decision maker may opt to solicit more information (in Act) and/or, if there is time, to wait and watch for those expectancies to be fulfilled in the developing situation. If they are, the associated COA is implemented. If not, another prototype may be activated. In any of these cases, there is no explicit problem solving, no evaluation nor modification of the COA. This process happens quickly and not necessarily under conscious control of the decision maker.

6.2.7.2. Familiarity - Moderate

406. Moderate familiarity suggests that no one prototypical schema in memory fits the situation immediately and exactly. Either several potentially-matching schemata or partial schemata are available in memory or there is only a general schema. In the first case, the understanding process triggers several viable prototypical situations and associated ones. If one is more strongly activated, it is used to guide further assessment using the cues for situation feature detection. The criteria in the prototype are used in determining the degree of match between prototype and perceived situation. If there is no strong contender, the potential prototypes may be compared (depending on time available), using degree of match, and the best chosen. If the situation is predictable, and time is short, the COA for the choice is implemented. In the case that the situation is dynamic or unpredictable, a check of expectancies in the developing situation is probably done as a further confirmation. The COA is mentally simulated against the developing situation, and if satisfactory, is implemented in Act. If the COA is not satisfactory a re-assessment of the situation may occur, (another prototype in the set is chosen) or the decision maker may act to gain more time.

407. It may be that the understand process triggers a partially-useful schema, producing the best match available, given the limited experience of the decision maker. The COA is evaluated (possibly using mental simulation) and it is discovered that it is only partially correct or complete. In this case, the COA is modified or further fleshed out to the detail necessary through Reasoning on the mental representation of the problem, and then re-evaluation. An alternative is that several schemata may fit different parts of the situation and their COAs may be selectively combined to produce a satisfactory COA for the situation.

408. The final case is when understanding produces a general prototypical description of the situation, but one that could have different instantiations (and thus different COA), depending on how the situation develops. If there is enough time, the decision maker may act to get more information on the expected sequence of events, or may mentally simulate the future sequence in hopes of establishing a likely one. The various potential situations and/or their associated COAs are compared on the basis of likelihood and viability. The decision maker may choose the most likely and

implement the COA or may try to maintain flexibility in COA to accommodate different contingencies.

409. In cases when a moderately familiar situation is presented, the process likely involves a degree of conscious problem solving. Because the COA has likely not been implemented in the past, the process also involves the generation of expectations about the effect of the COA, which are then confirmed or not through monitoring processes.

6.2.7.3. Familiarity - Low

410. In this case, understanding indicates a novel situation that is not matched by any experienced previously. The decision process is then one of developing a COA using the available information, and information from schemata on similar situations, or at least procedures that have been used to solve analogous problems. Reasoning processes work together with further understanding to develop the COA. The mental representation of the problem plays a large role in this kind of planning. The process may involve transforming the state of this representation using operators derived from weak or strong methods. Mental simulation of the potential COA is also important. If time is short, but the situation is predictable, then the decision maker may plan and implement a COA for a short period into the future, and confirm that the COA is satisfactory through Monitoring (assuming that the Act can be broken into several steps). If the situation is unpredictable, the decision maker may attempt to gain time on the decision to see how the situation will develop. When there is more time, it is possible to plan completely if the situation is predictable; if not, then several COAs may need to be developed to handle the contingencies.

6.2.7.4. Effects of other factors

411. The above discussion focused on the effect that familiarity might have on the decision process. We close with some brief comments on the effect of the remaining factors. The criticality (in terms of risk, costs, benefits) of the decision will bear on the effort put into finding or developing the COA. If the chance of loss is small, no matter what the decision, then the decision maker may not attempt to optimise. However, if the criticality of the decision is high (e.g., high risk, and/or high costs) then it is best to find a COA that minimises them, or at least falls below an acceptable level. Throughout, the time available to make the decision will be a factor in the overall strategy. Presumably the decision process will continue until either the decision maker is confident of the situation assessment and choice of COA (given the criticality) or time has run out.

412. In cases where the decision maker acts to gain more information it is necessary that the feedback from the environment be provided in time to support the process. This is less likely when the time for Acting is short. The overall strategy used for developing and implementing a COA may depend on whether there is more than one opportunity for action in the environment. If so, then a gradual or trial and error approach like a hedge-clipping strategy may be possible and more desirable than committing an untried COA to a high risk, unfamiliar situation.

413. In some cases the goals of the decision maker are strict and the constraints are tight. In other cases, they are less stringent. In the latter case, constraints that were at one stage in the process regarded as well-fixed might be loosened, especially if the amount of effort required to solve them is high and the risk/payoff is low.

This page has been left blank intentionally

6.3. COGNITIVE TASK ANALYSIS

6.3.1. Introduction
6.3.2. What is Cognitive Task Analysis?
6.3.3. Cognitive Model Generation
6.3.4. Knowledge Elicitation
6.3.4.1. Approaches to KE
6.3.4.2. Declarative Knowledge
6.3.4.3. Procedural Knowledge
6.3.4.4. Strategic Knowledge
6.3.4.5. KE Techniques

6.3.1. Introduction

414. The perspective that one selects for analysis will determine, to an extent, what will emerge in the specification of system requirements. Choosing a particular perspective means accepting an incomplete description of the complexity of actual task performance. A cognitive perspective will consider different aspects of behaviour than, for instance, a sociological or organisational one. (Of course, for decision making tasks it would be unacceptable not to consider the cognitive perspective because cognitive processes are at the heart of decision making.) In COADE we suggest that the analyst should take several perspectives in describing behaviour. Thus, in addition to a purely cognitive perspective, the analysis phase must take a behavioural perspective that accounts for the observable behaviours. COADE groups these other kinds of analyses under the term Task Analysis and Supporting Analyses.

415. It is important to position the cognitive processes that result from cognitive analysis in the larger context of the role of the human in the system. Thus Task Analysis (TA) is a precursor to identifying cognitive issues. It provides the temporal structure in which the cognitive performance operates and it gives a specification of the goal structure for the tasks. TA specifies the behavioural activities, aims and performance criteria. The subsequent Performance Analysis assists in the selection of the tasks that are the most critical. Critical tasks are those that have a large impact on the performance of the overall system. Cognitive analysis of these tasks has the highest likelihood of contributing to improvement of system performance.

6.3.2. What is Cognitive Task Analysis?

416. Cognitive task analysis (CTA) seeks to describe, in cognitive terms, how goals and tasks are accomplished. The value of cognitive task analysis is that it focuses on identification of the cognitive basis of behaviour and its limitations or, positively stated, the opportunities for support. CTA is concerned with:

a) the knowledge required for the task and the relationship among, and organisation of important concepts;
b) the mental operations for retrieval, storage, transformation, integration, and modelling of information;
c) the meta-cognitive processes that control cognitive effort and attention;
d) cognitive skill development and progression of knowledge structures from novice to expert.

417. CTA is critical in the design of any system in which the human has the responsibility for attaining the goals, since it provides information necessary for deciding which aspects of performance need to be supported, and how best to support them. For example, simply knowing that

a Navy decision maker will take action against a suspicious radar contact is not sufficient for determining how to provide support for that decision. A decision aid designer would need to know how the decision maker arrived at the decision, what cues triggered the thought processes, which schemata were activated, how information was perceived and transformed, and how the outcome was evaluated. In addition, the designer would like to know why and where the decision maker needs support.

418. A distinction should be made between the techniques or methods for Cognitive Task Analysis and the approaches or methodologies. The techniques include tools like questionnaires, interviews, critical decision elicitation. There are a variety of methodologies (organised sets of methods and techniques) that have been proposed to carry out CTA (e.g. Card, Moran & Newell, 1983 [GOMS]; Diaper, 1989; Grant & Mayes, 1991; Hollnagel. 1991; Johnson & Johnson, 1989 [KAT]; Klein, Orasanu, Calderwood & Zsambok, 1993; Payne & Green, 1989; Rasmussen, 1986; Rouse, 1991; Ryder & Redding, 1993; Schaafstal & Schraagen, 1992; Walsh, 1989 [ATOM]) The differences between approaches reflect different views of cognition and the lack of firmly established theoretical principles for analysis. Redding (1989) notes that methodologies for cognitive task analysis are limited and fairly ad hoc. Merkelbach and Schraagen (1994) found that CTA approaches were diverse and difficult to organise into a common framework.

419. The Naturalistic Decision Making (NDM) framework proposed by Klein (1993) and others has a focus similar to COADE. NDM concerns itself with how people make decisions in actual situations, in context (see also Section 6.1). Naturalistic tasks have features that make it difficult use results from 'clean' and simple laboratory tasks (see also Section 6.2.7 concerning situational factors). Some of these characteristics are:

a) time pressure
b) ill-defined goals
c) dynamic conditioning and shifting goals
d) inadequate information
e) cue learning
f) experienced decision makers
g) team coordination
h) poorly defined procedures
i) high stakes.

420. Despite their differences, the aim of each methodology is to capture the relevant cognitive requirements of the tasks. But for the methodology to work well adequate techniques and methods for tapping cognitive processes are needed. An obvious difficulty in the acquisition of cognitive information is the fact that cognitive processes and their content are not directly observable; they must be inferred by the analyst. This is complicated by the fact that many researchers believe that as expertise develops, knowledge becomes "compiled" and thus less consciously accessible to the individual. This means that experts may have difficulty in verbalising their reasoning or thought processes. Consequently, a variety of 'indirect' techniques, so-called <u>knowledge elicitation techniques</u>, have been developed to tap an expert's knowledge (e.g., Redding, 1989; Rouse & Morris, 1986). These techniques comprise interview and observation techniques, and modelling techniques.

421. Knowledge elicitation techniques can be grouped into three categories on the basis of the kind of information they elicit: <u>semantic nets</u> (declarative knowledge), <u>task-action mappings</u> (procedural knowledge), and <u>strategic knowledge</u>. For example, several techniques exist for deriving semantic nets, networks that describe the major concepts in a domain and how these are semantically related. Multidimensional scaling (MDS) is an example of such a technique. MDS uses concept ratings to determine the relationship of each concept to all other concepts, and then determines the dimensions in space that describe these relationships. MDS describes the conceptual structure in a different way from the original concept ratings. A variety of techniques have been developed to tap the more procedural aspects of cognitive performance. For example, there are interviewing techniques that are structured to yield task-action mapping models (Coury, Motte & Seiford, 1991). These models break cognitive tasks down into decision flow diagrams, that describe specific action sequences (including goals and subgoals) used for accomplishing the task. The final category of CTA methodology models how decisions are made in terms of decision strategies, cues, goals and expectancies. For example, the Critical Decision Method (Crandall & Klein, 1990) queries experts to

determine the decision making strategies they employed in response to non-routine events in their past.

422. Unfortunately, many of the CTA techniques developed to date have been devised in absence of well established theories of cognition, rendering their utility questionable. Moreover, applications in the past have often overlooked the purpose for collecting the data, failing to link particular methods with the ultimate use of the data. There are thus a variety of CTA techniques that differ greatly in how data are gathered, the form of the output, and the uses to which resulting data can be put. This makes it difficult for the prospective decision aid designer to select and utilise a CTA technique effectively. Further, it is often the case that the output of CTA (i.e., the cognitive terminology used to describe the task) is not rooted properly in cognition. The COADE framework addresses this pervasive problem by casting CTA in terms of the cognitive concepts delineated in Section 6.2. Thus, in COADE, CTA is used to determine the knowledge structures and processes (i.e., schemata) that characterise effective task performance. Once these categories of information are specified for a task, it is then possible to apply cognitive performance analysis techniques (see Section 6.5) in order to identify the aspects of performance most in need of aiding.

423. The goal in CTA is the generation of a cognitive model of a task, which is a specification of the knowledge required for a task and its organisation, together with the mental operations, mental models, and meta-cognitive processes involved. The model also specifies how these cognitive components interact with information in the world. The model (or set of submodels) is subsequently assessed for potential problems or opportunities during Cognitive Performance Analysis. The output of the latter analysis is a set of cognitive requirements for the task. The generation of cognitive requirements, a process in which cognitive task information and limitations are integrated and transformed into a format that is useful for designing a system, is crucial for systems design. The information used to generate the cognitive model is gathered using knowledge elicitation techniques that allow the analyst to describe in detail the content and structure of knowledge required to complete a task.

6.3.3. Cognitive Model Generation

424. The notion of a "model" ranges from abstract, formal, mathematical models to concrete, tangible, models such as mockups. A model is a representation of entities and their relationships in situations. The purpose of a cognitive model is to represent the relationships among the significant components of the cognitive system in order to describe, explain, and predict the cognitive behaviour. The cognitive model produced from analysis should structure the different aspects of cognition in such a way that its functioning corresponds with the observed facts, and practical aiding questions on cognitive demands and likely errors can be addressed.

425. The generation of a cognitive model for a specific decision-making task that is undertaken during ANALYSE will, in general, involve two steps. The first step is to adopt a generic model that is suitable for the task situation, if available. This model can come from theories of decision making or can be one of the more general models of cognition. (Models for decision making tasks are discussed in Section 6.1. A generic model applicable in C2 is given in Section 6.2. Situational factors that influence C2 decision making are presented in Section 6.2.7) A generic model can guide the analyst to critical decision making processes. The second step is to make the model task-specific, that is, fill in what specific knowledge is used and which processes are involved in a particular task.

6.3.4. Knowledge Elicitation

426. Knowledge acquisition or elicitation has been defined as the process of collecting knowledge from the decision makers and expressing it in the form of facts and rules (Chignell & Peterson, 1988). Generally, knowledge elicitation has two aspects: 1) psychological (i.e., the process of extracting the knowledge from the decision maker) and 2) computational (i.e., converting the knowledge into a usable structure). As noted earlier, a variety of techniques have been developed to tap a decision makers's knowledge, each with differing procedures and purposes. Indeed, one of the greatest challenges in eliciting knowledge is selecting a technique that is compatible with the

analyst's goal, and that will yield information that is valuable to the analyst's purpose.

427. Knowledge elicitation is used in COADE to document aspects of expert cognitive performance. That is, using a general cognitive model as a starting point (e.g., the knowledge representation model in Section 6.2), knowledge elicitation techniques are used to identify which types of inputs trigger or contribute to the decision making; what knowledge is invoked in making a decision and the structure of this knowledge; what type of processes or transformations occur using the knowledge structures; and what the output of the entire process (i.e., decision) turns out to be. The following sections describe how knowledge elicitation (KE) techniques can be used to generate the information necessary to specify the cognitive model of the task.

6.3.4.1. Approaches to KE

428. It is useful to distinguish KE techniques on the basis of the type of knowledge being elicited. Generally, three types of knowledge are considered to be descriptive of the decision maker's performance: declarative, procedural and strategic knowledge. Each of these types of knowledge contributes to expert performance in a different manner, and can be used as a means to classify KE techniques.

6.3.4.2. Declarative Knowledge

429. Declarative knowledge describes the facts, rules, concepts and attributes contained in a domain, and delineates the relationship among them. Declarative knowledge describes the content of knowledge employed by a decision maker in accomplishing a task. It corresponds to the "knowledge structure" portion of mentioned above. From the section on cognitive concepts (6.2), declarative knowledge under the COADE framework can be considered to include:

a) cues
b) criteria
c) expectations
d) goals
e) concepts
f) attributes
g) categories
h) constraints
i) relationships
j) conditions

430. These types of knowledge comprise one portion of the schemata that are hypothesised to be involved in expert decision making. From the designer's standpoint, it is necessary to specify the declarative knowledge associated with a task in order to then determine how knowledge is used in decision making, and to identify problems or limitations in schemata. Section 6.4.3 (Cognitive Limitations) provides more information on the manner in which schema-based processing may be limited. For example, decision making errors may occur because schemata are not properly formed; that is, inappropriate conditions may be embedded in the declarative part of the schema.

431. KE techniques will also assist in the compilation of the knowledge base that will underlie an aid that may be developed.

6.3.4.3. Procedural Knowledge

432. A second type of knowledge is procedural knowledge. This is knowledge regarding the steps, procedures, transformations and operations applied to knowledge in reaching a decision. In terms of schema theory as discussed in Section 6.2, procedural knowledge comprises the portion of the schema that describes what actions must be taken. Specifically, it includes:

a) rules
b) actions and action sequences
c) operators
d) heuristics
e) strategies
f) plans

g) procedures
h) processes

433. Procedural knowledge describes how declarative knowledge is used (or transformed) in order to reach a decision. According to schema theory, procedural knowledge is contained in schemata, so that it is retrieved in conjunction with declarative knowledge about that portion of the task. From a design standpoint, it is important to understand how decision makers use schemata (i.e., declarative knowledge with associated procedures), and how this processing may be flawed. For example, Section 6.4.3 (Cognitive Limitations) describes several types of errors that may be associated with inaccurate or incorrect procedural knowledge in a schema. For example, an execution error may involve a situation where the correct schema is activated, but an error occurs in the procedure (e.g., a computational error). Another type of procedural error is associated with reasoning, where a decision maker may apply strategies inconsistently.

6.3.4.4. Strategic Knowledge

434. A final type of knowledge important for the cognitive model and subsequent decision aid design is strategic knowledge. Strategic knowledge encompasses declarative and procedural knowledge. It involves the application of knowledge in the problem solving context; in particular, knowledge of the context in which procedures should be implemented, actions to be taken if a procedure fails, and how to respond if necessary information is absent. This type of knowledge is associated most closely with meta-cognitive processes described in Sections 6.2.4. Understanding how strategic knowledge is employed in decision making is crucial to a full description of cognitive task performance. For example, the meta-cognitive processes associated with strategic knowledge have to do with:

a) external monitoring of the demands or constraints in a situation
b) internal monitoring of the capabilities, beliefs and values applied to the problem
c) regulation or control of cognitive processes

435. In addition, strategic knowledge is useful for describing how declarative and procedural knowledge are combined in decision making. Transformations associated with these functions include the following schema-based concepts:

a) search
b) compare
c) choose
d) create
e) evaluate
f) modify
g) join, merge, link
h) order
i) group, categorise
j) abstract (chunk)
k) generalise
l) classify

436. These processes may be related to errors in decision making itself (i.e., making inappropriate evaluations) or to learning errors (see Section 6.4.3.). For example, decision makers may link concepts incorrectly, or make inappropriate abstractions or chunking.

6.3.4.5. KE Techniques

437. Taken together, the types of knowledge described here (declarative, procedural and strategic) can provide a rich description of a task in cognitive terms. The challenge for analysts is to apply appropriate KE techniques so as to provide an adequate understanding of how the task is performed so that the likely cognitive limitations can be specified. Unfortunately, as noted earlier, little attention has been paid in past work to the type of knowledge being elicited using various techniques. In an effort to ameliorate this situation, Table 6.1 provides a comprehensive description of current KE techniques.

438. Table 6.1 is organised according to the type of knowledge they tap (declarative,

procedural, or strategic). The table is split in two parts for readability.

a) The information in Table 6.1 Part 1 is as follows:
- **Technique** indicates the name of the technique
  Description provides a brief description of the technique
  Representation describes the representational format of the knowledge that is collected
  Sources provides references to the literature that give more detail on its development, procedures, validity and utility

b) The information in Table 6.1 Part 2 is as follows:
  Technique indicates the name of the technique
  Strength describes the strengths of the technique
  Limitations describes the limitations of the technique
  Subprocess in decision making/application indicates which schema-based concepts are tapped by the technique (see Section 6.2)
  Knowledge Type/Nature indicates whether the technique is direct (i.e., probes the expert for information by explicitly asking how he/she performs) or indirect (i.e., collects data without explicitly asking the expert). This column also indicates the type of knowledge tapped (i.e., declarative, procedural or strategic).

Table 6.1. Part I. Knowledge Elicitation Techniques
Technique, Description, Representation, Sources

DECLARATIVE			
Technique	**Description**	**Representation**	**Sources**
Card Sorting (general)	Knowledge engineer obtains set of concepts that broadly cover the domain (derived from glossary, text, or gleaned from introductory tutorial talk), then transfers each concept to a card. Expert sorts concept cards into common groups/ functions according to similarity (expert creates sorting criteria). These groups themselves can then be grouped until eventually a hierarchy is formed.	Hierarchical Cluster Diagram The end result is a tree of related concepts, with the bottom level holding basic components of the domain and progressing through different levels of abstraction to higher order concepts relating them. EX: Decomposing the technical domain of a central heating system into a conceptual organisation of subsystems.	(Gammock, 1987; Gammock & Young, 1985; McDonald, Dearholt, Paap & Schvaneveldt, 1986) (McDonald et al., 1986)
Cognosis (software)	Expert utilises software to identify concepts, problem solving, and tasks used to achieve domain objectives. Data used to create a conceptual graph.	Conceptual Graph	(Woodward, 1990)
Data Flow Modelling	Expert interviewed. Knowledge engineer draws data flow diagram using data gathered from interview. Expert verifies diagram.	Data Flow Diagram defines the processes which are required to be a part of the explicit knowledge base; the 'data' or 'knowledge' which exists within the knowledge base.	(Swaffield & Knight, 1990) (See Gane & Sarson 1977)
Document Analysis	Knowledge engineer translates information from a document into a conceptual graph. Propositions are translated into nodes and arcs of the conceptual graph.	Conceptual Graph	(Gordon, Schmierer & Gill, 1993)
Entity Relationship Modelling	Expert interviewed. Knowledge engineer draws entity relationship diagram using data gathered from interview. Expert verifies diagram.	Entity-Relationship Diagram	(Swaffield & Knight, 1990)

Entity Life Cycle Modelling	Expert interviewed. Knowledge engineer draws entity life cycle diagram using data gathered from interview. Expert verifies diagram.	Entity Life Cycle Diagrams Represents the allowable status changes of an entity and the events which cause those changes.	(Swaffield & Knight, 1990)
Interviewing (general)	Interviewing is the most familiar method of elicitation. In a fairly simple manner, it generates quickly a lot of knowledge that indicates the terminology and main components of the domain.	Varies	(Evans, 1988; Gammock & Young, 1985; Gordon et al., 1993; Grasser & Gordon, 1991; Visser & Morais, 1991)
Laddered Grids	Elicitors question the expert. Domain concepts and relations graphed.	Rules: Graphs of nodes and labelled arcs.	(Shadbolt & Burton, 1989)
Object Oriented Modelling (software)	Expert fills in computer forms detailing objects and events. Data collected includes scripts, types, aspects, relations, and attributes. Network of objects created.	Network of objects (types, aspects, attributes).	(Riekert, 1991)
Proximity analysis, (analysis by Pathfinder)	Once data are gathered, computer creates network representation of domain concepts, indicates meaningful links, and assigns weights. Gives core structure of domain.	Network Structure Pathfinder focuses on local relationships between concepts..	(Gammock, 1987)
Ranking Augmented Conceptual Ranking (ACR), Unshuffle Shellsort (adapted)	Ranking is a scaling technique that produces an ordering of the objects of interest. This ordering can then be converted into scale values using one of a number of techniques (see (Chignell & Peterson, 1988).	Conceptual Ranking	(Chignell & Peterson, 1988)
Ratings analysed by Pathfinder	Expert rates similarity between concept pairs. Algorithm creates network diagram of concept similarity distance.	Link-Weighted Networks	(Schvanevelt, Durso, Goldsmith, Breen, Cooke, Taucker, et al. 1985)
Ratings analysed by Multidimensional scaling (MDS)	Expert rates similarity between concept pairs. MDS algorithm assigns set of spatial coordinates for each concept. MDS considers the relationship of each concept to all other concepts and places the concepts along dimensions of space in a way that reflects these relations (MDS summarises data into a spatial configuration). Expert identifies dimensions of MDS graph.	Spatial Structure MDS focuses on global information about the concept space. In particular, successful identification of the dimensions of the space supplies information about the conceptual structure that cannot be gleaned from the original ratings nor from other scaling techniques (Schvanevelt et al., 1985)	(Schvanevelt et al., 1985)
Repertory Grid (general)	Method of deriving object descriptions Expert makes comparisons among groups of selections (typically triadic). These comparisons are used to identify attributes. A repertory grid is then constructed with attributes as rows and selections as columns. Using a rating scale, the expert rates the match between each selection and attribute pair on the grid (Chignell & Peterson, 1988).	Repertory Grids Can be analysed with cluster analysis, pathfinder, or multidimensional scaling.	(Evans, 1988; Gammock & Young, 1985; Gardner, 1990; McCloskey, Geiwitz & Kornell, 1991; Mitchell, 1987; Mitta, 1989)
Semantic Nets (Software)	Expert interacts with software (KNACK) to build a semantic net. Data collected includes relationships among objects.	Semantic Net	(Atkinson, 1990)

PROCEDURAL			
Technique	**Description**	**Representation**	**Sources**
Interviewing includes: - Backward Thinking - Concept Mapping - Constrained Processing - Free Generation - Decision rule elicitation - Picture Probes - Structured Interviews - Teachback	Variations of basic interview. include, working backwards through problem, drawing concept map, expert solving problem in limited time period, showing expert photographs depicting system in a number of states and asking questions, expert describes procedure to interviewer and interviewer teaches it back to expert,	Rules Concept Map (schematic representation of relationships among the task's components). Goal Hierarchy	(Andrus, 1988; Bainbridge, 1979; Barnard, Wilson & MacLean, 1986; Chignell & Peterson, 1988; Grasser & Gordon, 1991; Johnson & Johnson, 1987; McNeese & Zaff, 1991; Shadbolt & Burton, 1989); Thordsen & Klein, 1991; Woodward, 1990)
Discourse Analysis (observation)	Expert helps user, conversation is recorded. Conversation transcript analysed for tasks and subtasks. Data converted into a taxonomy.	Taxonomy of tasks/subtasks or functions.	(Belkin & Brooks, 1988)
IDEF Modelling	Structured Analysis Tool Expert interviewed. Interview team creates functional decomposition diagram (IDEF). Expert validates.	IDEF Model A highly structured syntax which facilitates functional decomposition. IDEF model provides a systems perspective and thus contributes to the identification of information required.	(McNeese & Zaff, 1991) Note: expert said, " This is not how I go about thinking about what I do" (p 1184)
Model-Based Reasoning (software)	Expert or knowledge engineer utilises software package to create a schematic diagram of the domain. Data collected includes characteristics of the system's main components and connections among components.	Schematic Diagram	(Hashemi, 1990)
Observation (induction)	Knowledge engineer observes expert perform the task (expert does not talk aloud). Observation used to identify underlying rules of task performance. Rules added to conceptual graph as goal/rule structure.	Goal and Rule Structure (Conceptual Graph)	(Gordon, 1993)
Petri Nets	Expert interviewed. Expert converts flow chart into a petri net. Data collected includes states and transitions (nodes and branches), constraints and conditions on sequence of transitions, tokens (information, data, conditions) passed from state to state. Help to analyse the dynamic behaviour of the modelled systems at various levels of abstraction and also represent the flow of information and resources.	A functional Task Net describes step by step how a task is executed including the interaction among team members. schematics of procedural steps, verbal lists of procedural steps, rules (inductive) underlying task performance, and creation of a goal rule conceptual graph.	(Coovert, Cannon-Bowers, & Salas, 1990; Hura, 1987; Weingaertner & Lewis, 1988)
Protocol Analysis Analysis of familiar tasks Static simulation	Expert works through a problem. Expert "thinks aloud" and explains reasoning for decisions made. Behaviour (verbal and nonverbal) recorded. Data converted to a set of productions that transforms one solution state to the next.	Production System Rules	(Andrus, 1988; Bainbridge, 1979; Gammock & Young, 1985; Hoffman, 1989; Shadbolt & Burton, 1989; Visser & Morais, 1991; See also Leplat & Bisseret 1965; Duncan & Shephard 1975)

Questionnaire	Expert performs task. Expert completes questionnaire on behaviours performed.	Behaviour description: - Sequence of task actions. - Cause and effect relationships.	(Bainbridge, 1979)
Schema Based Knowledge Elicitation (Software)	Expert draws a time-event diagram of an activity that describes how the activity is accomplished, and thus creates a template representing a class of situations. Data collected includes participants, events, and relationships between events.	Schema Template for a group of similar activities, formatted in Pascal data records, used by situation assessment inference systems.	(Noble, 1989)
Task Action Mapping	Expert identifies goals, subgoals, and actions needed to complete each task element of a decision flow diagram. Decision flow diagram then translated into a rule-based representation, with each goal and subgoal broken into action sequence.	Decision Flow Diagram translated into more rule-based representation; goals & subgoals decomposed in action sequences; purpose of action sequence is to provide a procedural description of system specific actions required to accomplish the task.	(Coury et al., 1991)

STRATEGIC			
Technique	**Description**	**Representation**	**Sources**
Critical Decision Method (Interview)	Interview of expert to identify non-routine events that challenged expertise and events to which expertise made a significant difference. 2) time-line of event constructed. 3) Key points further probed.	Goals considered during incident; options generated, evaluated & chosen; cue utilisation; contextual elements; situation assessment factors specific to particular decisions, decision strategies;	(Thordsen & Klein, 1991) See also (Klein, Calderwood & MacGregor, 1989)
Critical Decision Method (CDM) (Interview)	Semi-structured interview utilising specific probes designed to elicit particular type of information. Data examined for perceptual cues, judgement details, and decision strategy details that are not generally captured with traditional reporting methods.	Decision Strategies	(Crandall & Klein, 1990)
Decision Graph (Software)	Expert uses graphical interface to create a decision graph.	Decision Graph (Tree)	(Rodi, Pierce & Dalton, 1989)
Goal Directed Analysis	Technique is designed to map the relationship between parts, how the parts work, how evidence testifies the state of these parts and how each can change as a function of the state of the domain. Knowledge gathered from multiple sources including interviews, documents, observations and simulations. Goal means network created.	Goal-means Network (functional interrelationship): Structure of domain task in terms of goals, relationships between goals, and the means to achieve goals.	(Woods & Hollnagel, 1987)

Policy Capturing (Ratings)	General procedure designed to describe statistically the unique information processing strategies of individual raters. Expert rates performance profiles. Regression analysis used to objectively demonstrate the expert's combinations and weights of the information. Scores on the separate elements/ dimensions used to compute relative importance.	Information Weights	(Hobson, & Gibson, 1983)
Policy Capturing (Ratings)	Policy capturing explicates the relative weights, functional forms, and the strategy for combining environmental information sources (cues) into a summary judgement (Hammond, Mumpower, Smith, 1977; Dougherty, Ebert & Callender, 1986) Expert rates a hypothetical case or person. Data identifies judgement structures and tendencies, relative weights, and functional forms. Multiple regression equation uses information cue values to predict decision judgements. Multiple regression equation weights reveal unique decision policies.	Information weights	(Dougherty et al., 1986)
Storyboarding (Interview)	Storyboarding prototyping provides a medium within which to transform the language based representations inherent in concept mapping and IDEF modelling into an object-oriented design. Allows expert to experience the prototype design.	A prototype display design based on Task/Action Mapping. Display design: Expert illustrates on paper what he/she needs on the display surface during the performance of the mission; expert identifies what is needed on a display to support a decision point.	(McNeese & Zaff, 1991)
Task Action Mapping	Expert identifies goals, subgoals, and actions needed to complete each task element of a decision flow diagram. Decision flow diagram then translated into a rule-based representation, with each goal and subgoal broken into action sequence.	Decision Flow Diagram translated into more rule-based representation; goals & subgoals decomposed in action sequences; purpose of action sequence is to provide a procedural description of system specific actions required to accomplish the task.	(Coury et al., 1991)
User Needs Analysis	Approach to design of information system that identifies the information needs of the user, reveals the reasoning process & decision strategies employed by users to make decisions, and represents those processes and information requirements in such a way as to enhance system development. User needs analyses and current management practices used to create models of decision process and data flow diagrams for specific tasks.	Decision Process Diagrams	(Coury et al., 1991)

Table 6.1. Part 2. Knowledge Elicitation Techniques
Technique, Strength, Limitations, Subprocesses in decision making/Application, Knowledge type/Nature

DECLARATIVE				
Technique	Strength	Limitations	Subprocess in decision making/Application	Knowledge type /Nature
Card Sorting (general)	Gives structure to large concepts sets, easy to do (Converse & Kahler, 1992) appropriate for systems with natural hierarchical organisation (Tullis, 1985). Apart from detailed knowledge which experts bring to bear on specialised areas, experts are likely also to have a more global structuring of the domain. Concept/ card sorting helps identify this meta-knowledge (Gammock & Young, 1985).	Requires prep work to create concepts; requires knowledge engineer trained in interpretation; requires computer; hierarchy may be too restrictive; permits only one view per sort; some aspects may be distributed and lost (Converse & Kahler, 1992);	Knowledge Structures: concept / categories, goals, principles / values, relationships. ——— Applicable when a large set of concepts exists, which range across the whole domain, and which require a suitable structuring to become manageable.	Declarative ——— Indirect
Cognosis (software)	Can use this method to build a domain model before using specific knowledge acquisition tools.	Requires Cognosis software and computer hardware.	Knowledge Structures: principles / values, goals, schema, concepts / categories, relationships	Declarative (concept graph is taxonomic, spatial region hierarchy, or causal network); Procedural (concept graph is goal hierarchy) ——— Direct
Data Flow Modelling	Method defines boundary between knowledge that needs to be explicit and knowledge that doesn't.	Expert's task being modelled may not have a sequential flow, requires two knowledge engineers: one to interview, one to draw diagram (Swaffield & Knight, 1990); Requires training in data flow diagram methodology (Converse & Kahler, 1992)	Knowledge structures: goals, schema, concepts / categories, strategies, relationships.	Declarative (inputs and outputs); Procedural (processing flow of inputs to outputs); ——— Direct
Document Analysis	Method can detect missing information, inconsistent information, and ambiguous statements.		Knowledge Structures: goals, schema, concept / categories, rules	Declarative (conceptual graph is taxonomic, spatial region hierarchy, or causal network) Procedural (goal hierarchy) ——— Direct
Entity Relationship Modelling		Requires two knowledge engineers: one to interview, one to create diagram (Swaffield & Knight, 1990); Requires knowledge engineer trained in entity-relationship modelling methodology (Converse & Kahler, 1992).	Knowledge structures: goals, concepts / categories, relationships	Declarative ——— Direct

Entity Life Cycle Modelling		Difficult to represent inheritance through control relationships, Requires two knowledge engineers: one to interview, one to create diagrams (Swaffield & Knight, 1990); Requires knowledge engineer trained in entity life cycle modelling (Converse & Kahler, 1992)	Knowledge Structures: goals, schema, concepts / categories, rules, relationships	Declarative (allowable entity status changes); Procedural (events which cause those changes); Direct
Interviewing (general)		Knowledge may not be directly communicable in interview situations. Instead, it must be inferred using other techniques (Gammock & Young, 1985).	Knowledge structures: principles / values, goals, schema, concepts / categories, rules, strategies, relationships	Declarative Direct
Laddered Grids	Highly similar to interview format	Requires knowledge engineer trained in rule analysis (Converse & Kahler, 1992)	Knowledge structures: concept / categories, goals, relationships	Declarative Direct
Object Oriented Modelling (software)		Experts must be comfortable thinking in terms of objects, requires specific computer software and hardware (Converse & Kahler, 1992).	Knowledge structures: goals, schema, concept / categories, relationships	Declarative Direct
Proximity analysis, (analysis by Pathfinder)	Proximity analysis retains local structuring and thus complements the multidimensional scaling (Converse & Kahler, 1992) Pathfinder extracts the latent structure rather than transforming the data, and thus is better able to reflect psychological proximity on a pairwise basis.	May require arbitrary estimates as input data (Gammock, 1987); Requires computer, requires knowledge engineer trained in network interpretation (Converse & Kahler, 1992)	Knowledge structures: goals, rules, concepts / categories, schema, relationship	Declarative indirect
Ranking Augmented Conceptual Ranking -> (Chignell & Peterson, 1988); Unshuffle (Kagel, 1986); Shellsort (adapted) (Whaley, 1979)	(ACR), (Chignell & Patty, 1987); Unshuffle Shellsort (adapted)		Knowledge structures: conceptual / category, relationships	Declarative Indirect
Ratings analysed by Pathfinder	Pathfinder extracts latent structures. This better reflects the pairwise (local) psychological proximity than do data transformations (Converse & Kahler, 1992)	Global information not included (Schvanevelt et al., 1985); Requires advance identification of concepts, requires knowledge engineer trained in link-weighted networks, requires pathfinder software and computer (Converse & Kahler, 1992);	Knowledge structures: concept / category, relationships	Declarative Indirect
Ratings analysed by Multidimensional scaling (MDS)	MDS captures inter-concept global relationships. MDS creates a metric (distance between concepts in multi-dimensional space) that has useful applications (Gammock, 1987)	MDS can distort local distance relationships (within a concept pair), MDS requires expert interpretation, requires MDS algorithm (Schvanevelt et al., 1985); Requires advance identification of concepts (Converse & Kahler, 1992)	Knowledge structures: concept / category, relationships	Declarative Indirect

Repertory Grid (general)	Validated by utilising grid **analysis** results to predict performance on cognitive tasks within the domain; Statistical techniques identify hidden patterns, grid permits expert to compare his/her understanding to the analysis results to evaluate agreement level, experts may make adjustments for special cases; Appropriate when numerous closely related concepts require expertise to discriminate among them (Gammock & Young, 1985).	Memory drain on expert (Converse & Kahler, 1992) Little procedural knowledge provided (Evans, 1988)	Knowledge structures: concept/ categories, relationships ——— Repertory grid method appropriate when numerous closely related concepts require expertise to discriminate among them (Gammock & Young, 1985).	Declarative ——— Indirect
Semantic Nets (Software)	Domain expert not required to know any programming language, artificial intelligence schemas, rule semantics or other computer science abstractions; KNACK increased productivity of potential expert system users 30 times that of previous methods.	Requires KNACK software and compatible hardware (Converse & Kahler, 1992).	Knowledge structures: concepts / categories, relationships	Declarative ——— Direct

PROCEDURAL				
Technique	**Strength**	**Limitations**	**Subprocess in decision making/Application**	**Knowledge type /Nature**
Interviewing includes: - Backward Thinking - Concept Mapping - Constrained Processing - Free Generation - Decision rule elicitation - Picture Probes - Structured Interviews - Teachback	Could use to validate rules acquired from protocol analysis; Method requires little training time for both interviewer and expert, method considered highly effective by domain experts (Thordsen & Klein, 1991). Leads to identification of insights, decision elements, beliefs, and information requirements (McNeese & Zaff, 1991).	Requires knowledge engineer trained in interviewing, other general problems associated with interviewing, expert must be comfortable with thinking backwards (Converse & Kahler, 1992); Interview highly time consuming, may not elicit certain aspects of domain.	Knowledge structures: concepts / category, rules, relations, goals Working memory: assess / understand, evaluate, mental models Processes: monitor, assess / evaluate.	Procedural ——— Direct
Discourse Analysis (observation)		Requires inter-coder reliability, time consuming, subjects must consent to auditory taping ; experts may be unable to articulate problem solving expertise if domain tasks and goals are not well defined (Belkin & Brooks, 1988).	Knowledge structure: goals, concepts / categories, rules, relationships	Procedural and Declarative ——— Direct

IDEF Modelling	Successfully represents a functional decomposition of the expert's task.	Diagram may be visually complex; diagram cannot capture all dynamic aspects of task; structure of IDEF model induces expert to represent task in ways incompatible with his or her conceptual understanding of the task ; IDEF structure ambiguous: distinctions among components not clear cut; interpretation requires knowledge engineer trained in IDEF (McNeese & Zaff, 1991)	Knowledge structures: concept / categories, relationships, schema Processes: assess / understand, evaluate	Procedural (functional decomposition of processing stages); Declarative (identification of inputs, outputs, constraints, and processing mechanisms); ________ Direct
Model-Based Reasoning (software)	Model based reasoning advantages: Single model available for use by several analysis packages, systems and components can be viewed from different perspectives.	Requires PLEXSYS software, requires computer hardware to suit PLEXSYS software (Converse & Kahler, 1992)	Knowledge structure: concept/ category, relationship, schema	Declarative (information about behaviours of each object within the domain); Procedural (control sequence of each object, control of overall system); ________ Direct
Observation (induction)		Complex implicit knowledge may not be perceived with inductive analysis (Gordon, Schmierer & Gill, in press).	Knowledge structure: Goals, relationship, rules, schema Working memory: schema, mental models, perceived situation Processes: monitor / sense, assess / understand, evaluate, act	Procedural ________ Indirect
Petri Nets	Models can be verified through comparisons of teams and validated through comparisons of other net configurations (Coovert et al., 1990); Petri nets can model a system in a hierarchical manner and represent relationships between processes in the system (Hura, 1987).		Knowledge Structures: Schemas, rules, goals, concepts / categories, relationships, strategies	Procedural ________ Direct
Protocol Analysis Analysis of familiar tasks Static simulation	The main purpose of protocol analysis is to identify structures and patterns rather than simply to look at contents (Byrne, 1983; Evans, 1988) Includes more than experts are able to explicitly describe during a problem solving situation; permits inference of knowledge expert uses but that expert does not verbalise and may not be aware of.	Requires knowledge engineer trained in verbal reporting and protocol analysis, other problems associated with verbal protocols (Converse & Kahler, 1992).	Knowledge structures: rules, schema, strategy, relationships Working memory: mental model, perceived situation Processes: search / reason, evaluate, act	Factual propositions (Declarative); Procedural propositions (Procedural) ________ Direct

Questionnaire		Questionnaire may not be able to identify all situations, requires knowledge trained in survey design, other problems associated with questionnaires (Converse & Kahler, 1992).	Knowledge structure: Rules, schema, relationships, strategy Processes: evaluate	Declarative ——— Direct
Schema Based Knowledge Elicitation (Software)	Schema based knowledge elicitation procedures help an expert communicate his/her schemata to a computer. These schemata are represented as data structures for knowledge-based expert systems that infer the meaning of situations from a pattern of observables. Schema represent knowledge in a flexible way and reflects human tolerance for vagueness, imprecision, and quasi-inconsistencies (Rumelhart, 1977; Noble, 1989)	Experts uncertain how to partition activity into events, expert uncomfortable creating single template for several contexts, experts with different situational assessment specialities represent different aspects of same data ; Requires project planning software and specific hardware (Noble, 1989); Method based on assumption that knowledge organised as schemas; Not all types of knowledge required for situation assessment represented by project planning software (Noble, 1989).	Knowledge structure: schema, goals, rules, relationships, strategy Processes: assess / understand, evaluate Working memory: mental model, perceived situation	Procedural ——— Direct
Task Action Mapping	Action sequences provide the level of detail necessary to specify the interactions that must occur at the system level for the user's task to be accomplished.	Requires knowledge engineer trained in creating task action mappings (Converse & Kahler, 1992).	Knowledge structure: rules, schema, goals, strategies Processes: assess / understand, evaluate	Procedural (familiar, rule-based situations); Strategic (novel, knowledge-based situations) ——— Direct

STRATEGIC				
Technique	**Strength**	**Limitations**	**Subprocess in decision making/Application**	**Knowledge type /Nature**
Critical Decision Method (Interview)	Effective in eliciting deep, difficult-to-articulate tacit knowledge that separates experts from novices.	Requires knowledge engineer trained in interviewing; relies a certain degree on expert's memory.	Working memory: schema, mental model, perceived situation Processes: assess / understand, evaluate, meta-cognition	Strategic (decision strategies, critical cues, situation assessment, goals/intent, expectancies, mental simulation strategies, and improvisation) ——— Direct
Critical Decision Method (CDM) (Interview)	Yields information of richer variety, specificity, and quantity than typically available in experts' verbal reports (Crandall, 1989).	Requires knowledge engineer trained in interviewing; reliance on recollection ignores human's mediocre recollection; other problems associated with interview (Converse & Kahler, 1992)	As Above	Strategic (goals, options, cue utilisation, contextual elements, situation assessment factors). ——— Direct

Decision Graph (Software)	Knowledge engineer does not need to be present during knowledge acquisition, no translation of knowledge required.	Cannot handle novel situations (Rodi et al., 1989); requires computer and software, expert must be familiar with graphic interface (Converse & Kahler, 1992).		Strategic ——— Direct
Goal Directed Analysis	Networks characterise types of problems solved in a domain and how human performance affects those problems; networks provide a framework to discover problems that can arise and kinds of information processing requirements; can identify points in the process where multiple interpretations and errors may occur;	Results may depend on knowledge source (Converse & Kahler, 1992)	Knowledge structure: goals, relationships, strategies, schema Processes: assess / understand, evaluate	Procedural (knowledge from documents, observations, interviews); Strategic (knowledge from specialists and simulations); ——— Direct
Policy Capturing (Ratings)	Extracts expert's policy (i.e., decision making, strategy) using actual decisions as input.	Context may influence ratings (e.g., administrative versus research setting), format varies between numerical and verbal descriptions which can influence raters, potential unstable and falsely inflated R2, number stimuli relative to number dimensions too low (Hobson & Gibson, 1983); requires computer and software, requires knowledge engineer trained in policy capturing, raters judge appropriateness of policies but raters have been shown to have little insight into the policies (Converse & Kahler, 1992)	Knowledge structures: relationships Processes: monitor/sense, evaluate	Strategic (element importance and applications) ——— Indirect
Policy Capturing (Ratings)	As Above	Raters evaluate hypothetical cases (Dougherty et al., 1986); requires computer and software, requires knowledge engineer trained in policy capturing, raters judge appropriateness of policies but raters have been shown to have little insight into the policies (Converse & Kahler, 1992)	As Above	Strategic ——— Indirect
Storyboarding (Interview)	Storyboarding gives expert the opportunity to translate his/her conceptual knowledge and expertise into a representation and design prototype which could be perceptually experienced by other viewers of the storyboard.	Appropriate for visually-oriented tasks; specific to display design.	Knowledge structures: concept / categories, relationships, schema Working memory: mental model Processes: monitor/sense, search/reason	Strategic (information requirements and display element relationship to task actions) ——— Direct

Task Action Mapping	Action sequences provide the level of detail necessary to specify the interactions that must occur at the system level for the user's task to be accomplished.	Requires knowledge engineer trained in creating task action mappings (Converse & Kahler, 1992).	Knowledge structures: schema, rules, goals Working memory: mental model Processes: assess / understand, evaluate	Procedural (familiar, rule-based situations); Strategic (novel, knowledge-based situations) ———— Direct
User Needs Analysis	User needs analysis identifies data and information required for the topic, determines availability of data, and reveals functional and organisational relationships among users. User needs analysis combined with cognitive modelling provides an extremely useful method for capturing and incorporating the decision processes of users in the design of information systems. (continued)->	Requires knowledge engineer trained in creating decision flow diagrams (Converse & Kahler, 1992). ———— ->continued When based on user needs analysis, cognitive models provide a user centred approach to developing decision models for information processing systems. The cognitive models structure and organise decision strategies, and produce decision models for a system that is congruent w/ the user's model of the decision problem.	Knowledge structures: concept / category, goals, schema, rules, relationships, strategies Working memory: mental models, perceived situation Processes: monitor/sense, assess / understand, evaluate, meta-cognition	Procedural (familiar situations); Strategic (novel situations); ———— Direct

This page has been left blank intentionally

6.4. COGNITIVE PERFORMANCE ANALYSIS

6.4.1. Implications of Cognitive Task Analysis for Performance Analysis
6.4.2. Approaches to Cognitive Performance Analysis
 6.4.2.1. Structural Constraints
 6.4.2.2. Cognitive Workload
 6.4.2.3. Comparison to Ideal
 6.4.2.4. Amount of Deliberation
 6.4.2.5. Case-based Findings
 6.4.2.6. Cognitive "Principles"
6.4.3. Cognitive Limitations
 6.4.3.1. Schema Limitations
 6.4.3.2. Learning Errors
 6.4.3.3. Errors in Knowledge Representation
 6.4.3.4. Basic Processing Errors
 6.4.3.5. Limitations of Metacognition
6.4.4. Analysis of Cognitive Limitations

439. Cognitive performance analysis is one of several analyses in COADE. It involves the examination of the cognitive model of processes and structures produced in the cognitive task analysis. The cognitive performance analysis should identify what aspects of cognition are critical for successful performance. Cognitive performance analysis is the means to judge the acceptability of cognitive performance for the purpose of determining the most critical or most likely conditions for improvement. The cognitive performance analysis has a heavy emphasis on cognitive limitations. These are statements about the cognitive cause for behavioural performance errors. Analysis of the limitations is based on assertions about the frequency and criticality of the behavioural and cognitive errors and the feasibility of aiding them. The cognitive performance analysis results in the identification and specification of the cognitive requirements.

440. Cognitive requirements are statements about what are the most critical cognitive processes, knowledge structures, and controlling strategies for a task. Cognitive limitations or errors are related because they distinguish aspects of thinking that are prone to be undesirable in some way. The intent of cognitive performance analysis is to assess what the undesirable aspects of thinking might be. The limitations may be purely speculative and hypothetical, because cognition is especially difficult to measure or because the task is a prospective one for future systems. In any case, the search for cognitive limitations is valuable to ensure that the root cause of problems are identified so they can be overcome. Analysis of cognitive errors leads to increased understanding which in turn leads to opportunities for improvement.

441. The previous section on knowledge elicitation techniques (6.3.4) describes ways of obtaining information about cognitive aspects of specific domains. However, typical knowledge elicitation techniques by themselves do not provide enough guidance to identify limitations or requirements. Techniques for knowledge assessment and engineering lag behind those for elicitation and acquisition. This section explores ways that cognitive performance can be analysed to produce cognitive requirements.

6.4.1. Implications of Cognitive Task Analysis for Performance Analysis

442. There are several approaches to cognitive task analysis (Grant & Mayes, 1991). Some of these support COADE's cognitive performance analysis, but the cognitive task approaches are not

especially suited for addressing how well people think (see Section 3.2 of Workshop Summary). Redding (1989) notes that techniques for cognitive task analysis are fairly ad hoc. Redding indicates that the techniques have several elements in common: ways to measure abilities, task components, differences between conceptual and procedural knowledge, differences between novices and experts, and steps to progress from one knowledge state to another. Although there are similarities among cognitive analysis methods, they produce a variety of results, do not share specific procedures, and most do not have special techniques to identify cognitive limitations or strengths.

443. The lack of firmly established theoretical principles for analysis and different views of cognition have led to different approaches. (Grant & Mayes, 1991) proposed an approach that focuses on information flow and usage. Roth, Woods, and Pople (1988) reported a method that identifies the difficult aspects of a problem, the demands on cognitive ability, and the ways people respond to the demands that incur errors. Methods like these have a general orientation on behaviour or cognition, but do not provide detailed guidance.

444. Hopson and Zachary (1982) outlined a method with information categories. The analysis is called the summary tabulation of aiding requirements (STAR) (also see Zachary, 1986; Zachary, 1988; Zaklad et al., 1986). They identified types of information that they felt would be desirable for analysis of decision problems. The STAR technique is principally a list of the following information categories:

a) Decision situation.
b) Task dynamics.
c) Situational objective.
d) Value criteria.
e) Underlying process.
f) Information environment (input, output, parameters).
g) Intermediate reasoning and analysis steps.
h) Representation.
i) Required judgements.

445. The existing cognitive analysis techniques describe cognitive processes or memory for specific task domains to some extent. However, the techniques do not provide much guidance on how to assess the extent that cognition contributes to task performance or how it is susceptible to limitations.

6.4.2. Approaches to Cognitive Performance Analysis

446. Grant and Mayes (1991) describe existing methods for cognitive analysis as either decomposing a task to match the human's cognitive structure, modelling the units and structure of cognitive processes and resources, or dealing with specific findings of cognition (like expert - novice distinctions, and Rasmussen's levels of behaviour, 1982). Several approaches are possible to use for the identification of cognitive limitations. These include the following concepts:

a) Structural constraints.
b) Cognitive workload.
c) Comparison to some specification of ideal.
d) Amount of deliberation.
e) Case-based findings.
f) Cognitive principles.

447. There is overlap among these approaches. Developers of cognitive analysis methods have combined the different concepts since each offers something of value. Each approach will be discussed briefly to review their desirable qualities and limitations for use in COADE. The cognitive principles approach is based on cognitive theory, empirical findings, and expert-novice differences and is incorporated for use in COADE.

6.4.2.1. Structural Constraints

448. Cognitive limitations can be defined as structural constraints on memory size, processing rate, and attention limits. Zachary (1986, 1988) based his technique on structural constraints and two

other problems in processing. He identified five general information processing limitations:

a) Working memory has a limited capacity from three to nine chunks.
b) Reasoning requires a minimum amount of time for each operation (at least .1 second).
c) Recall from long-term memory is unreliable, depending on the recency, frequency, and relationship with what is currently in working memory.
d) Mental calculations are difficult and prone to error; people prefer approximate reasoning over calculations.
e) People are not as accurate in their visual imagery as they believe.

449. His premise was that these limitations lead to a small set of cognitive problems:

a) Process prediction.
b) Choice criteria and combining attributes.
c) Retrieving information.
d) Reasoning.
e) Visualising or relating data to a mental model.
f) Inaccurate quantitative judgements.

450. And in turn these problems related to six categories of decision support system functionality.

a) Process models.
b) Value models.
c) Information management tools
d) Automated analysis and reasoning techniques.
e) Representation aids.
f) Judgement refining and amplification techniques.

451. Each decision support category contained a number of techniques for aiding. Selection of an aiding technique is based on the underlying model or characteristics of the problem. Zachary did not provide a theory of or explicit description of cognition to guide an analyst's application of STAR or the aiding solutions. His lists indicate that there are limits on attention and activation of knowledge and that higher cognitive processes require more cognitive effort. While Zachary's lists of limitations and cognitive problems are useful as a starting point, specific domain should be explored in greater depth to identify cognitive requirements that lead to more specific aiding solutions.

6.4.2.2. Cognitive Workload

452. Cognitive workload offers the possibility of applying communications principles to thinking processes. The basic operational notion of workload is one of capacity for processing (similar to channel capacity of a communications system). One characteristic of a capacity approach is that people have limited capacity at any one time. There is only so much sensory data that can be perceived and attended to. There is a maximum amount of information that can be retrieved from memory and some limited capacity for processing. These limits are stated in terms of the load that there is on cognition.

453. One approach to analysing workload is to measure spare capacity. The analysis starts with a level of task difficulty that can be accommodated by the subject population. Task requirements are added and performance is observed to notice any change. As loading increases, there should be some point when either primary or secondary task performance becomes noticeably worse. This can be useful for inferring or predicting the capacity limit. The allocation of attention is another consideration of cognitive workload. The question is whether the most important aspect of a task is attended to at the right time. Stress, fatigue, motivation, and effort all are attributes which impact capacity.

454. A cognitive workload approach is not completely suitable because it provides an incomplete explanation of why performance capacity is exceeded. It does not capture underlying reasons for exceeding memory, processing, and meta-cognition. A cognitive workload approach reveals something about the limits of cognition, but does not indicate specific problems of knowledge content, rules, or meta-cognition that might lead directly to an aid to correct a specific problem.

6.4.2.3. Comparison to Ideal

455. Another approach can be based on the position that perfect rationality is desirable. This approach takes the position that optimal decisions will be made if decision makers are totally logical, place optimal weights on information, use optimal criteria, and use optimal algorithms to arrive at the best decision. Analysis would be done by comparing observed performance to the ideal. The major problem is that the ideal may be some artificial statement based on some inappropriate theory of what is ideal. It is commonly known that people do not follow principles of complete rationality in their decision making (e.g., Klein, 1993; Simon, 1956). This is partly due to the effects of uncertain and incomplete information; situations are not completely predictable and deterministic. It is not likely that many complex tasks in command and control have optimal components that simply need to be discovered and applied. The "comparison to ideal" approach presumes that an ideal can be defined and should be used prescriptively for improving decision making.

456. Another way of considering a rationality approach has been the identification of decision biases. Biases have been described as heuristics that lead to incorrect outcomes. They are typically assessed relative to a normative standard. Hogarth (1987) has identified four types of biases and 31 sources of individual biases. These and additional biases are briefly described in Table 6.2. The literature identifying biases typically assumes a rationale decision maker and is based on laboratory studies that have substantial differences from everyday tasks.

Table 6.2. Overview of biases (adapted from Hogarth, 1987).

Bias	Source	Description
Acquisition	Availability	The likelihood of something happening may be judged by how easily examples of it come to mind.
	Selective perception	What is not or cannot be perceived is not used. Prior knowledge imposes structure on a task.
	Frequency	People use observed frequency instead of observed relative frequency to assess events.
	Illusory correlation	People can misjudge the frequency with which two events occur and their degree of association.
	Data presentation	Judgement is affected by how information is structured: order, sequential vs. intact, qualitative vs. quantitative, missing information, and context.
	Framing	Whether information is presented as gains or losses, positives or negatives will affect peoples' goals and assessments of outcomes.
Processing	Inconsistency	People do not apply a strategy consistently.
	Conservatism	People fail to update assessments when new information is received.
	Non-linear extrapolation	Exponential processes and joint probabilities are under- or over-estimated.
	Habit	Choices are based on prior satisfaction.
	Anchoring and adjustment	People use a cue value as an anchor and adjust for a new cue. Insufficient adjustment leads to underestimation.
	Conjunction	People tend to think that descriptions that are more elaborate are associated with a higher likelihood.
	Representativeness	People tend to assess that some object or event is more likely to generate some other object or event if the two are similar. People tend to equate similarity of surface characteristics with probability instead of using sample size and base rates.
	Law of small numbers	People tend to expect that a small sample will be representative of random chance.
	Justifiability	People tend to believe that an inappropriate decision is justified if it is based on a rational rule.
	Sunk costs	People tend to concentrate on previous course (nonproductive venture with heavy investment) instead of ignoring that and focusing on future costs and benefits.
	Misconception of regression	Regression toward the mean is often ignored.
	'Best guess'	People tend to ignore that information can be unreliable and uncertain. People tend to think they know more than they do and that what they don't know is unimportant.

	Consistency	Consistency of information, without an associated increase in accuracy, leads people to be more confident.
Output	Question format	The chosen or required ways of making judgements affects the outcome, e.g., people are willing to give more to remove a risk than to acquire an equal reduction of risk.
	Wishful thinking	The probability of an outcome is higher when it is based on desire rather than knowledge.
	Illusion of control	People tend to feel that due to their skill they have more control over events than is actually the case.
Feedback	Misperception of chance ("gambler's fallacy")	Observation of unexpected similar chance outcomes leads to an expectation .that the likelihood of events not recently observed increase.
	Attribution	People attribute undesirable outcomes to bad luck and desirable ones to skill.
	Hindsight	Knowledge of events shifts or recreates the memory of prior predictions, so that people think they are better than they are and fail to make adjustments.
	Outcome irrelevant learning	People think that observable outcomes provide information that also indicates something about events that did not occur or were not observed.
	Logical fallacies in recall	Inability to recall details leads to 'logical' reconstruction which can be inaccurate.

457. Recently Gigerenzer (1991) has actively challenged many of the biases indicating that they are not in fact violations of probability theory but artifacts of narrow conceptual views. The biases can be made to disappear in at least two ways: (a) by distinguishing between single case judgements and frequencies of judgements over time and (b) by accounting for previous knowledge about a problem domain.

458. Analysts considering the use of biases need to consider that people follow natural strategies that are powerful and potentially more efficient than so-called optimal strategies. The application of biases is limited because they do not conform to an overall model or any cohesive set of theories. Besides people are driven by principles of cognitive economy. The strengths people adopt to deal with time-constrained, uncertain situations (e.g., abstractions, generalisations, shortcuts, heuristics) can also be the liabilities they have in other conditions. The specific task contexts, individual knowledge structures, and processes must be considered to understand the implications of performance. Since biases may actually be desirable heuristics, biases cannot be simply attributed to performance and targeted for correction by some decision aid purporting probabilistic magic. The underlying cognitive characteristics must be explored further.

6.4.2.4. Amount of Deliberation

459. Also important is Rasmussen's (1986) framework of knowledge, rule, and skill levels. Rasmussen's individual levels can be conceived as three areas on a continuum of the amount of deliberation. The continuum or gradient can be thought of as a composite of different identifiable characteristics. The characteristics might reasonably include how much knowledge is typical, how abstract the knowledge is, how general or specialised the rules are, and how automatic or deliberate the processes are. Rasmussen's knowledge-based reasoning is a careful, deliberate use of knowledge with only the most basic or general pre-existing schemata. Anderson's related level is cognitive thinking. Rasmussen's second level is rule-based thinking which occurs when existing schemata are instantiated. Anderson's second level is associative thinking. Rasmussen's third level is called skill-level behaviour that is automatic. The thinker is unaware of the activation of schemata. This might be the level which is most related to intuitive thinking. Anderson's related level is autonomous behaviour.

460. Reason (1990) has characterised errors that are related to each of Rasmussen's levels in a generic error-modelling system (GEMS). Knowledge-level behaviour has errors based on bounded rationality and a complex, ambiguous problem space. Existing rules are not directly available to the problem as it is presented. The solution is to transform the problem into conditions for which existing rules apply or to generate (induce) new rules or responses that might apply. Rule-based errors are related to the misapplication of good rules or the application of flawed rules. Skill-based errors fall into the meta-cognitive category when there is too little or too much attention paid to the processing.

461. Reason (1990) provides a description of various errors that fall into each of these categories (see Section 5.1.2.3 -Table 5.5). He reports that generally people can detect all but about one fourth of their own errors, and their ability to make corrections is the lowest at the knowledge-based level.

462. Reason's system is a possible technique for the analyst because it relates to three fairly apparent levels of deliberation. If the analyst knows the decision maker's familiarity with a problem the processing level can be inferred. Or the processing level can be observed directly. Once the processing level is known, then errors or failure modes might be predicted or distinguished using GEMS. An analyst can use the errors or failure modes for observation and classification of behaviour or in a predictive fashion to eliminate anticipated errors. GEMS does not completely suit COADE because it is not tied to a very detailed model of cognition. Skill, rule, and knowledge-based processing are fairly gross categorisations. Complex decision making like that which occurs in C2 tasks, corresponds most closely to knowledge-based processing. Failure modes are made up of biases and one other subcategory. Because of the reservations noted above about biases and a fairly small list of other knowledge failures, the GEMS was assessed as not being sufficient for COADE's cognitive performance analysis.

6.4.2.5. Case-based Findings

463. Another approach to cognitive analysis is to use existing or similar cases to assess cognition. Previous knowledge elicitation is a means that may have already produced findings that are applicable to a new problem. Elicitation techniques are also useful for collecting new information for a problem (see Section 6.3.4.5). There may be known desirable and undesirable aspects of cognition which have been previously identified. The discipline of expertise can provide information about perceived or real success. Knowledge about expertise may be based on characteristics of expert processing or specific knowledge that experts have.

464. One assumption is that if nonexperts are provided with the expert-like knowledge or trained in the expert processes, then their performance should progress to higher levels. There are two points that need to be highlighted which provide a balancing view. For one, it may not be appropriate to accelerate expertise or to design conditions to impose aspects of expertise on a non-expert. Expertise may not be able to be accelerated, and the non-experts may be incapable of dealing with the different knowledge or processes. The other point is that experts do not necessarily always perform at optimal levels. Their own performance is subject to predictable and random errors. An expert's more abstracted rules may not generalise to some conditions, and because of the expert's confidence he or she may not recognise the misapplication. These considerations are mentioned so that expert models are not blindly applied to create expert-based systems.

6.4.2.6. Cognitive "Principles"

465. A related, but broader approach to identify cognitive limitations is to use emerging general knowledge about cognition. Appendix A on Cognitive Concepts summarises current beliefs about cognition. The concepts provide a description of the "principles" of cognition. If taken as principles or rules of cognition, then the summary implies an outline for categorising various errors. This is the approach taken below to identify possible cognitive limitations.

466. Appendix A on Cognitive Concepts highlights several areas of principles upon which the development of cognitive limitations is based. Consideration of schema-based thinking led to limitations of schemata as knowledge structures and the processes involved in instantiation. Limitations associated with the act of learning can be identified during the formation of schemata and other processes. Knowledge structures, reasoning processing, and meta-cognition provided three more areas in which limitations could be projected. Limitations were identified based on the principles of cognition and other taxonomies and lists of errors.

6.4.3. Cognitive Limitations

467. The following list of cognitive limitations makes up most of the approach for cognitive performance analysis. This method can be said to be "loose;" it is not a step-by-step method for compiling data to deduce specific limitations about cognition. Strict methods have been attempted

before for task and cognitive analyses. The strict methods are characterised by the elements of information which should be filled out. Specifying interesting elements of information a priori is useful, but typically that relies ultimately upon the analyst's insight to identify limitations and requirements.

468. The purpose of this cognitive performance analysis approach is similar in that it is based on the premise that cognition is the proper level of analysis, but it goes beyond an outline description by specifying types of cognitive limitations. The instances of cognitive limitations are derived from cognitive principles. Undoubtedly there are other possible errors that are not on the list. The descriptions serve as a general checklist to stimulate search and recognition. The cognitive principles can be separately judged for accuracy and usefulness.

6.4.3.1. Schema Limitations

469. Schemata are specialised knowledge structures which combine procedures and cues for when those procedures should apply. Examination of potential schema-based errors provides a way to unify the other principles of cognition. Limitations can be tied to three schema processes: instantiation, formulation, and procedural execution.

470. Instantiation. Limitations on schemata primarily involve instantiation. Instantiation involves searching and recognising schemata which may fit a given situation. It involves the filling of the schema slots for schema that already exist for a particular situation. It can also be construed to include the adaptation of several existing schema for a new situation. Rasmussen's (1986) level of skill-, rule-, and knowledge-based thinking and the section on decision processes (6.2) provide an indication of different amounts of deliberation involved in instantiation. Quickly or gradually (depending on the familiarity of the situation) the schemata will be tested for feasibility of application to a specific situation. If any schemata does not appear to fit, schemata with relevant elements will be combined or modified for the apparently novel situation. The following statements capture ways in which the instantiation process can go awry. Finding the right schema may the source of the problem.

a) Incorrect fitting of data to the schema.
b) Incorrect filling of slots with guesses instead of data (similar to failing to search or construct a relevant model).
c) Inappropriate use of subordinate schema (over-specialisation).
d) Inappropriate use of superior schema (over-generalisation).
e) Error in accretion: an experience is incorrectly assessed as another.
f) Error in tuning: incorrect elaboration and refinement of concepts.
g) Inappropriate use of most common schema; forced to fit the situation (similar to a "habit" bias).
h) Several schemata are triggered, but wrong one is picked.
i) Schemata are confused. Common elements represented at a higher level incorrectly called into script.
j) Existing schemata are relied upon too heavily, reluctance to generate a specialised schema.

471. Formulation. Schemata can also be limited by errors embedded when a schema is initially formed or when [illegible] schema are used when a new schema should be developed. Errors during formulation or l[illegible]ng occur when the wrong conditions or rules are embedded in the schema. Schema formulation problems can also occur when aspects are omitted, or schemata simply not formed.

a) Inappropriate conditions embedded in the declarative part of the schema.
b) Inappropriate rules or responses embedded in the procedural part of the schema.
c) Key parts of rule conditions are omitted. (Compounding is a process of developing a new rule through simplification by intersection of conditions of two rules.)
d) Nonstandard elements of schemata have not been stored as pointers or tags, so are unavailable to form different schemata or to differentiate among existing ones.

e) Schemata are not formulated when appropriate to have them.

472. The instantiation limitations above can be viewed also as detailed explanations of failed schema formulation. For example, over-specialisation or over-generalisation could occur during formulation. Experiences can be incorrectly classified or learned as another so confusion results during instantiation or execution.

473. Execution. Another type of schema limitation occurs when the execution of the procedure fails to produce the anticipated result. This may be due to processing or meta-cognitive limitations. The correct schema is activated, but an error occurs in procedure (e.g., computational error).

6.4.3.2. Learning Errors

474. Learning errors associated with schema-based thinking are described above in formulation. Taking a more general perspective on thinking processes leads to the statements below on limitations in learning. Learning errors can occur because of problems in classification, feedback, or rules.

Classification. Broadened understandings are a critical part of learning. The failure to recognise common and dissimilar attributes of concepts, objects, and events is an important limitation. Too many distinctions or constant changes in classification impede thinking as well.

a) Links among concepts are not made or made incorrectly.
b) Important attributes are left out of classifications when they are formed.
c) Memory structures are excessively reorganised when new experiences and repetitions occur.

Feedback. A decision maker's attention to feedback is used to test and adjust expectations.

d) Failure to attend to feedback will prevent important information from being used to strengthen existing knowledge, differentiating among knowledge, and forming new understandings and rules.
e) Insensitive to feedback (related to feedback biases).

Rules. The ability to induce rules from ambiguous cues is a characteristic of advanced cognition.

f) The failure to induce new and more applicable rules is a sign of difficulty in learning.
g) Rules or regularities are not generalised to induce new rules.

6.4.3.3. Errors in Knowledge Representation

475. Schema knowledge structures have already been addressed as part of the instantiation errors. Other knowledge structures include concepts and mental models. General limitations associated with other knowledge structures include lacking knowledge, poor organisation of stored knowledge, and poor models of current situation.

a) Rudimentary or insufficient knowledge and relationships, including poor goals, values, or "world" knowledge.
b) Poor organisation of knowledge.
c) Inability to use or retrieve appropriate representations.
d) Inappropriate crossed memories.
e) Poor integration of knowledge or poor representation for a particular state.

6.4.3.4. Basic Processing Errors

476. Understanding, generalisation, and reasoning are three general types of cognitive processes. These processes are used to categorise various processing errors.

Understanding. There are several terms for processes which lead to the

understanding of the state of the world and the state of a problem. Monitoring, recognising, interpreting, assessing, and comprehending can all be involved in determining what is happening and what ought to or can be done. Errors of understanding can be decomposed into some of the following statements. Instantiation errors described under schema are similar.

a) Poor encoding or representation of the situation and its meaning.
b) Ignoring important classification attributes.
c) Failure to recognise salient features and critical relationships in a problem.
d) Failure to consider implications of models identified in the search.

Generalisation. Errors in generalisation, abstraction, or chunking can be of two general types. Either knowledge is combined that should not be, or there is knowledge that should be chunked and it is not. When knowledge is at the inappropriate level of specificity, reasoning can also occur at the wrong level of abstraction. Thinking can be too concrete when principles are not identified and applied, or it can be too general when inexact and vague models are applied. When too general the focus can be on unimportant aspects, or general rules can be formed when continued use of specific instances would be more appropriate. These generalisation errors are related to the over-generalisation and over-specialisation problems described under schema-based thinking.

e) Missing or inappropriate abstractions or chunking.
f) Incorrect normalisation transformation to an event that was not most likely or typical.
g) Thinking occurs at wrong level of abstraction.
h) Too few abstractions are used, too few multiple relations.

Reasoning. Reasoning makes up a general cluster of processing limitations. Depending on the specific problem, the reasoning processes can differ greatly. Problems with strategy selection, uncertainty, projections, and evaluation are prominent examples of reasoning difficulties.

i) Inappropriate strategy selection (incorrect schema).
j) Inconsistent application of a strategy.
k) Inappropriate relational or logical reasoning.
l) Inability to hold in mind various possibilities.
m) Poor trade-offs about importance.
n) Ignoring uncertainty rather than trying to resolve it.
o) Failure to project ahead.
p) Inadequate search for counterexamples.
q) Inappropriate use of analogical reasoning.
r) Failure to critique, check for consistency, validity of assumptions.
s) Failure to de-conflict ambiguous information.

6.4.3.5. Limitations of Meta-cognition

477. Limitations of meta-cognition (the regulation of thought) depend to a large extent on the view of its role in cognition. The following statements provide a fairly complete view of meta-cognition. These controlling processes are subject to limitations if they are not evoked or if they are poorly done. The following list corresponds to a proposed three part classification of meta-cognition: (1) external monitoring of the demands or constraints in a situation, (2) internal monitoring of the capabilities, beliefs, and values that one has to apply to a problem, and (3) regulation or control of cognitive processes.

External monitoring. External monitoring involves how one thinks about the influence of external conditions on his or her own thought processes.

a) Failure to recognise that a situation requires something to be done.
b) Failure to gauge difficulty of a problem.

Internal monitoring. Internal monitoring involves thinking about one's own capabilities, shortcomings, beliefs, and values to choose how to deal with a problem.

c) Failure to assess likelihood of knowing.
d) Failure to monitor actions, evaluate one's strategy.
e) Failure to organise thoughts.

Regulation. Regulation is the process of controlling one's thought processes, based on external and internal monitoring.

f) Inability to allocate attention and cognitive resources.
g) Poor use of time.
h) Failure to set goals.
i) Inability to synchronise processes.
j) Inability to control actions, revise one's strategy.
k) Planning is overly opportunistic; lacks adequate control.

478. These meta-cognitive limitations are also useful for considering potential disconnects in a group process. For example, a group can fail to set or agree on goals, they may not intentionally allocate tasks among participants, and they may not bother keeping focused and using time wisely. (The organisational analysis activity in the COADE framework has other implications for important group aspects.)

6.4.4. Analysis of Cognitive Limitations

479. Analysis of cognitive limitations should produce hypotheses about what types of limitations are responsible for the behavioural errors identified in the performance analysis. The implied requirement is for the analysis to determine which limitation is responsible, so a design requirement can be generated to address that cognitive limitation or statement of a cognitive requirement. The field of applied cognition is not advanced sufficiently to produce that information with absolute rigor. The analysis of cognitive limitations is a craft and will remain so until considerable additional experience is obtained in the field as a whole. The following produces some general guidance in how the list of cognitive limitations might be used and an brief example.

480. Given a list of potential cognitive limitations, the operative issue is which limitation applies to a performance problem in a given task. There is likely to be several possible limitations responsible for poor performance. Several limitations may have the same root cause so it can be useful to consider the relationships (symptom-cause-effect) among different errors. When multiple errors are anticipated or substantiated they can be differentiated along several dimensions. The first two ways involve the likelihood of self-detection and the frequency of occurrence. The third way of distinction deals with the criticality of the limitation, and the fourth way looks ahead to the feasibility of remedying the limitation.

481. The analyst should judge whether limitations that are self-detected are important for aiding. If self-detected and appropriate corrective measures are taken by the decision maker, then aiding may not be required. Limitations that occur frequently, but that are not self-corrected should be considered by the analyst because a single cognitively-centred aid may correct a large proportion of undesirable behaviour.

482. Criticality of the limitation is perhaps the most important consideration the analyst must make. When limitations occur frequently and are critical, then the analysts and developers can easily focus their efforts here. If a limitation or the situation that might prompt a limitation rarely occurs but the result is critical, then the analyst and developer need to consider whether a cognitive requirement and associated solution are warranted. Some errors may be frequent and critical, but the feasibility of over coming the undesirable effects is unknown. In these instances it is probably best to identify a cognitive requirement, but to realise the amount of effort and risk involved with targeting aids to address it. The frequency, criticality, and feasibility of correction all contribute to the assessment of the potential payoff of addressing specific limitations.

483. Example. To be useful the identification of cognitive limitations need to help produce the cognitive requirements and in turn the design requirements. An example of dealing with minefields on a battlefield is discussed. The example illustrates the sequence of analysis and conclusions that traditional aiding efforts may have taken and the additional benefit of cognitive analysis.

484. Keeping track of minefields has been a problem observed during Army command post training exercises. Minefields can be established by both enemy and friendly forces and can be installed by a variety of means. Over the course of a battle, it can be difficult to keep track of minefield locations, especially when battles are dynamic are go back and forth over the same terrain. Existing minefields are both potentially dangerous and useful to manoeuvre forces and planners working on future operations. Planners should be aware that the effects of battle actually alter the nature of terrain from what is depicted on standard tactical map sheets or earlier aerial images of the area. As the terrain is altered so is the mobility afforded by the terrain. It has been observed that planners sometimes forget about earlier minefields or disregard them when planning forces to manoeuvre in an area. An example statement of a performance problem at a task or performance level is that, 'the staff has failed to consider important information.'

485. Without consideration of the cognitive implications of the error, an analyst or developer might propose to build a decision aid to keep better track of minefield information and display it as a standard part of an electronic terrain and situation map. This may not be a useful approach. The information may not be available for display or may not have been reported reliably. If the planners have a rich schema or mental model of a dynamic battlefield, they ought to be aware that minefields (and other battlefield "clutter") are likely to be present. The planners should have anticipated the existence of minefields in the vicinity of previous engagements or mobility corridors and should have proactively sought the information or, lacking the time to confirm locations, to anticipate the effects of possible minefields on manoeuvres (e.g., general slowing of movement or canalising of engagement areas).

486. Not knowing the principles of minefields and not having sufficient schemata have different implications than not recognising that certain conditions exist. Not knowing suggests remedial instruction or an on-line help system, while not recognising certain cues exist (like, manoeuvring over a previous engagement area) suggests an information-based decision aid to represent critical information along with a "meta-cognitive" component to broaden considerations or to provide alerts.

487. Applying the COADE list of cognitive limitations an analyst might predict that the staff did not have sufficient knowledge about a specific battlefield instance, did not appropriately anticipate minefields, or did not know what action to take concerning minefields. Once predicted the analyst should try to determine through knowledge elicitation or other means what is the typical cognitive limitation.

488. Considering the possibilities, the analyst might attribute the problem to poor schema formulation. The decision maker's schemata for terrain were formulated without indication of an exception condition on previous battlefield clutter. Let's imagine that the analyst states this as a tentative cognitive limitation and then conducts knowledge elicitation to corroborate the assertion. In knowledge elicitation, the analyst finds that everyone he interviews does not immediately think about the effects of previous battlefield actions on terrain. When the analyst asks specifically about it, everyone does know its significance and specific ways in which old minefields need to be taken into account. Since everyone has a component in a schema they already know the information, so the problem is not one of learning. There is still the possibility that at the time of schema formulation that previous battlefield effects were not prominently tagged as an "exception." Another way to trace the limitation is to say that it is a failure in processing, a failure to recognise salient features and critical relationships in a problem. Another aspect of the limitation may be that the decision maker made a reasoning error in that he did not consider what assumptions were unstated. A related possibility is a meta-cognitive failure to allocate time to check assumptions. All these views can be used in statements of cognitive requirements. Further knowledge elicitation or evaluation could pinpoint the cause and indicate an even more specific cognitive requirement and possible design requirements. If all statements of limitation still stand, then solutions should try to address all aspects. If a full elaboration of the limitation cannot be made, then the cognitive requirement will be stated more generally.

This page has been left blank intentionally

6.5. PERFORMANCE AIDING STRATEGIES AND GUIDELINES

6.5.1. Aiding Approaches
6.5.2. COADE Performance Aiding Strategies
6.5.2.1. Control of Reasoning Processes (Meta-cognition)
6.5.2.2. Reasoning Processes (Cognition)
6.5.2.3. Information Handling
6.5.3. Guidelines for Designing Decision Aids
6.5.3.1. Guidelines Derived from Failed Decision Aids

489. Performance aiding or decision aiding can be done from quite varied approaches. A listing of various performance aiding strategies will be useful to assist requirements analysts and designers. The list can help to find an appropriate strategy for a specific application. A listing should provide sufficient exposure to a variety of techniques so that a wide range of concepts can be considered and designed in detail. A list of aiding strategies provides an approach complementary with the analysis to generate cognitive requirements. These strategies provide the building blocks for ways to satisfy cognitive requirements.

6.5.1. Aiding Approaches

490. Philosophies of aiding can be as numerous as there are different approaches for cognitive performance analysis (Section 6.4). The approaches discussed in that section included structural constraints, cognitive workload, comparison to ideal (normative) performance, amount of deliberation, case-based findings, and cognitive principles. Instead of attempting to identify cognitive limitations, the aiding strategies for design are targeted at overcoming limitations. There are analogous approaches for aiding, yet there are also different issues implied in aiding than in identifying requirements.

a) Should the focus be on aiding the task or on the underlying cognition used in the task?
b) Should the aid address easy, but time consuming tasks, or what is difficult?
c) Should the aid concentrate on frequent activities or on infrequent, but critical ones?
d) Should general purpose aids be developed that can assist with common elements of skill or develop aids that are narrowly focused with specialised knowledge?
e) Should limitations be prevented or should negative effects from the limitations be alleviated?
f) Should the aid provide assistance or change the nature of the task to bypass the human limitation?
g) Should cognitive limitations be replaced altogether with machine intelligence?
h) Should aids be based on expert capabilities or should aids be built to allow novices to do better at using their own capabilities?

491. COADE does not take a singular and absolute position in answering the questions raised above. These questions should be based largely on the analyses of task, performance, and cognitive limitations. The design of aids will also take into account the constraints of the development: what does the organisation expect and want in the way of decision aids, what trade-offs should be made between amount of change desired and acceptable levels of development risk, and how many resources can be invested in decision aids? The guidelines and lists of aiding strategies in COADE are presented to provide different perspectives for the analysts and designers to consider when dealing with a specific application.

492. Several researchers have addressed the issue of philosophies of aiding. Rouse (1991) laid out a general framework for philosophies of aiding. He proposed three philosophies of support. One goal is to inform those that are not aware; a second goal is to enable those that cannot do something to do it; and the third goal is to encourage those that do not want to do something to do it. Informing, enabling, and encouraging are goals that can be sought simultaneously in the development of aids.

493. Meister (1991) extends the approach of Murphy and Mitchell (1986) to contend that the architecture of an aid should be as close as possible to the structure of cognition (e.g., since human knowledge has some characteristics of hierarchical organisation then displays should be hierarchical). But there is no absolute reason that the aid should have the same characteristics as people. Taken to extremes this position would imply that the system would have the same limitations as people. So it is not imperative that the aid's structure mimic cognition. More importantly it should help overcome limitations and extend capabilities.

494. Weber and Coskunoglu (1990) argue for using optimal decision approaches in decision aids as long as they can be flexibly applied to changing problems and situations. Several researchers point out the difference between a normative or prescriptive approach and a naturalistic or personalised approach. Carrier and Wallace (1989) recommend pointing out the difference to the decision maker so that he or she is prompted to select the most appropriate approach. Cohen advanced the notion of combining personalised and prescriptive approaches into a single aid (Cohen, Bromage, Chinnis, Payne & Ulvila, 1982; Cohen, Leddo & Tolcott, 1989).

6.5.2. COADE Performance Aiding Strategies

495. To develop a listing or taxonomy of strategies several perspectives were considered. Previous lists of decision aids were reviewed (e.g., Accinelli, Robinette, & Jacobs, 1985; Andriole, 1989; Carter, Archer & Murray, 1988; Hopple, 1986; Payne, Braunstein, Ketchel & Pease, 1975; Phelps, Halpin & Johnson, 1981; Puckett, 1990; Ramsey, & Atwood, 1979; Rouse, 1991; Sprague & Carlson, 1982; Witus, Patton & Cherry, 1985; Zachary, 1988). None of these lists cover the complete range of how aids could enhance performance. In fact, an exhaustive listing will never be possible, because we will continue to develop new strategies as technologies advance and greater insight into performance and cognition is acquired. Several different clustering approaches were considered until the present taxonomy resulted. The COADE taxonomy was developed by clustering aiding concepts according to various cognitive characteristics. Focusing on cognitive goals and operations makes this listing unlike many of the previous ones, which were organised by technological, quantitative, or information processing concerns.

496. The cognitive goal-based taxonomy has three top-level categories of aiding (see Figure 6.5). The two initial categories were generated from a distinction between helping the decision maker apply a strategy better and helping the decision maker by providing a different strategy. The first, Control of Reasoning Processes, deals with concepts that all have to do with improving the application of a skill or reasoning. The second category, Reasoning Processes, is made up of concepts that target the content and procedures used in reasoning. The third category includes general information handling aids. Another way to view the categories is its division into three types of human processes: meta-cognition, cognition, and information management.

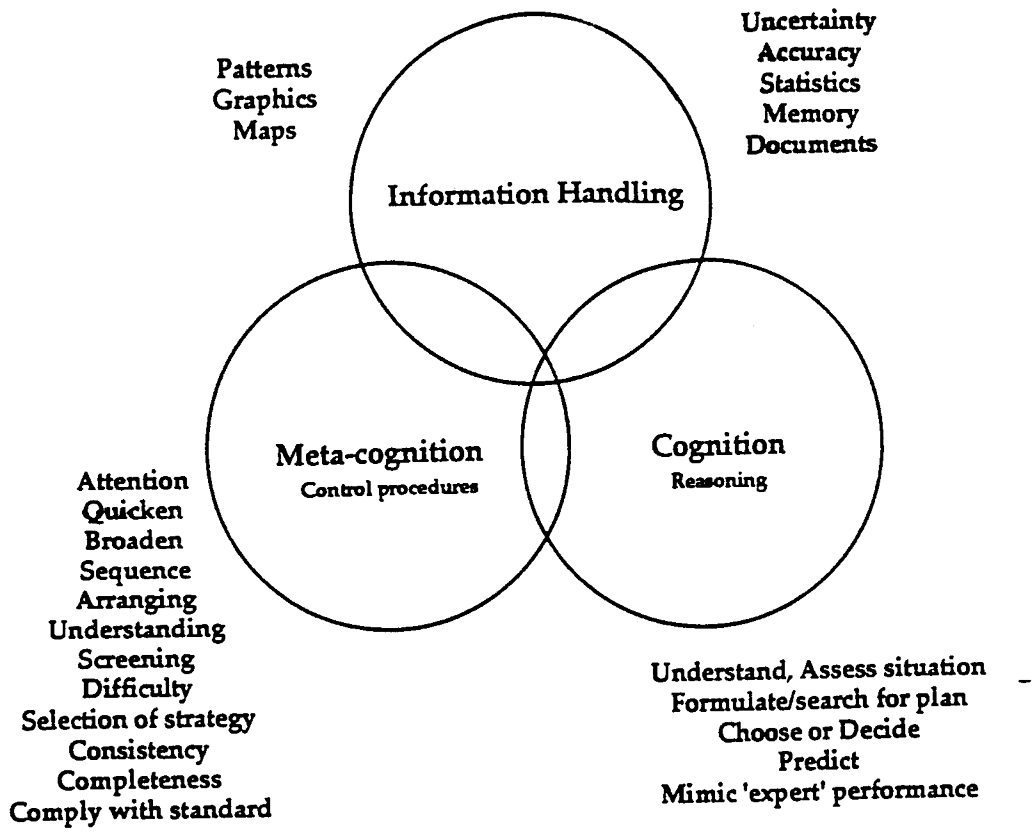

Figure 6.5. A Cognitive Goal-based Taxonomy of Performance Aiding Strategies.

497. These strategies are not mutually exclusive, nor are they hierarchical. The overlaps of the various strategies are not explicitly shown, but can be realised by comparing similar operations or imagining how different operations can be used together. There are six types of information contained in the columns of the Table 5.9 (See Section 5.2.1.2). Category refers to one of the three clusters of aiding strategies: Control of Reasoning Processes, Reasoning Processes, and Information Handling. "Goal" refers to what the strategy tries to achieve. The Basis of the Aiding Operation provides a description of how individual strategies work. Examples of typical strategies provide terms commonly associated with the concept. Strengths and limitations are given, that provide comments on practical issues of potential merit or difficulties in the development, utility, usability, or acceptance of the strategies. Some of these assessments are based on empirical findings and others are extrapolated from similar strategies.

498. To gain the most from the taxonomy, users should consider multiple goals. The list is not an exhaustive inventory of strategies or a list of intact strategies. The list provides various strategies that could be blended together to develop a hybrid application for particular cognitive requirements. Each of the perspectives offers a different way to view performance and could lead to different possible aiding strategies. Users should consider the merits of multiple strategies and determine how they can be combined to satisfy the identified cognitive requirements. The categories, goals, basis of operation, strengths, and limitations should provide a departure point for identifying additional instances of strategies and for developing new ones.

6.5.2.1. Control of Reasoning Processes (Meta-cognition)

499. This category includes concepts which address how cognitive processes can be assisted with a computer aid. The common attribute of the techniques is similar to what cognitive

psychologists call meta-cognition (i.e., awareness of one's own thought and directing and regulating mental activities). Concepts which are included here are ones that do not contain domain specific knowledge or rules of reasoning. Since they can be implemented without specific domain knowledge, they should be usable across most tasks. (A significant question about meta-cognitive aids is the extent that more conscious control of cognition could interfere with "automatic" or skilled performance or could increase the amount of time required. These unknowns need to be tested.)

500. **Attention.** An attention perspective can be implemented to monitor information, to filter information, to distinguish, classify, and fuse information, and to provide alerts accordingly. Attention aids can stimulate the users when they might tend to lose attention when information is too infrequent or when there is so much information that it becomes difficult to sort out that which is most important. Problems can occur when the alerting function does not apply to a specific instance or when the system takes some action autonomously without informing the operator.

501. **Quickening.** Increasing the speed of applying a strategy is one approach to aiding. Speed may allow more of the solution space to be explored or may allow perusal of outcomes from rapid trial and error evaluation. The aid may have the decision maker encode his or her decision procedure, so that the computer can run it rapidly. Quicker is not necessarily better. When there is sufficient time and an encountered problem is relatively unfamiliar, it is probably more critical to apply or develop an effective procedure.

502. **Broadening.** The aid might be designed to stimulate creativity in a procedure by prompting users to generate as many goals, constraints, criteria, or options as possible; by having the user use analogies to discover new approaches; or by other creative-enhancing approaches. Decomposition of a problem can be used to identify the elements of the problem and when recombined the problem can be viewed from a different, broader viewpoint. Creative-enhancing techniques do not guarantee better solutions, but do generally lead to an increase in considerations.

503. **Sequence.** The modification of how skills are applied might be as simple as having the user follow a different sequence of activities or to "backtrack" and check for consistency in previous steps of the solution. Many tasks are too complex or too dynamic to realise much benefit from rearranging the steps.

504. **Arranging.** A related strategy is to operate on the allocation and scheduling of cognitive (or group) resources for performing the task. The strategy should allow ongoing updates because of the complex nature of most decision tasks. Scheduling and allocation tools can assist in the application of an individual's time and mental resources. Similar concepts can be used for coordinating and managing joint work. One group example of this type of strategy is for generation, distribution, and control of orders. Aids can help a chief of staff, an executive officer, other staff, or the commander to program and manage the work to get orders done on time. The intended result is to facilitate the coordinated efforts of the multiple players. By making work processes more complete and efficient, more concentration can be directed at more complex thinking tasks and overall better performance can be achieved.

505. **Understanding.** Another meta-cognitive strategy is to encourage the decision maker to better understand critical aspects of a problem. One way is by providing a guide for procedures to develop the situation understanding (sometimes called the intelligence preparation of the battlefield). Another strategy is for the aid to prompt the user to see the problem differently. Modifying the structure of the problem, goals, resources, or attributes is a common method of solving "insight" problems, where a solution becomes apparent after seeing new relationships. Too much disruption could lead to indecision and reluctance to act.

506. **Screening.** Having a feeling of knowing or not knowing information is important of problem solvers. The self-monitoring judgement is useful for determining the effort required to assemble appropriate knowledge. A screening strategy could help the decision maker assess his or her own knowledge or the knowledge available in the computer. Screening does not indicate what should be done if knowledge is not readily available.

507. **Judging difficulty.** Similar to assessing the availability or adequacy of knowledge, is assessing the likelihood of solving the problem (has a similar problem been solved before?). Knowing

the difficulty of the problem can help determine the approach to the problem.

508. Strategy selection. Determining how a problem is to be solved is another important meta-cognitive skill. An aid can assist in the selection, by recommending a strategy based on predetermined criteria or by guiding a user in following a choice strategy (see below).

509. Consistency. The aid can try to "learn" what strategy is used by the decision maker (or from many decision makers or an expert decision maker) and enforce or remind the decision maker of that strategy during problem solution; this is often referred to as bootstrapping. For example, the aid might monitor the relative weights given to various attributes of options and capture the trend over several instances. If the system detects large deviations in weighting in a subsequent instance, then it can try to bring about change by alerting the decision maker to the anomaly, or it might go ahead and automatically apply the previous weighting strategy. The aid can also take control and impose a computer solution derived from the user or programmed at some prior time.

510. Completeness. A meta-cognitive approach to completeness is to stimulate or remind the decision maker to consider various factors. Procedural checklists or checklists of factors to consider can stimulate thoroughness. Pre-flight aircraft checklists serve as a memory aid to ensure that important information is checked. Similarly military concepts such as Clausewitz' principles of war and METT-T factors (mission, enemy, own troops, terrain and weather, and time available) can be embedded in planning aids to encourage completeness of considerations.

511. Standards. Decision aids can be used to track on-going performance. This strategy can perform or support an assessment function, depending on the sophistication of the aid. The resulting information can be used to provide feedback to the user to assist with his or her continued learning and maintenance of knowledge. Also comparison to standards or typical system faults can help in the detection and diagnosis of failures. In other cases, an aid following "standards" could be programmed to "lock-out" the user from making inappropriate inputs.

6.5.2.2. Reasoning Processes (Cognition)

512. This category includes concepts which aid in understanding, improving choices, formulating plans, predicting the future, and emulating experts. This category depends more on the specific content of domain knowledge and the specific reasoning processes used. The analysis phase of COADE should generate the information necessary to consider reasoning strategies in more detail than the descriptions given below.

513. Understanding. Thoroughly understanding the problem is critical in order to choose or develop a solution. An aid can help in the instantiation of schema or the recombination of existing schemata to fit the current situation. Aids can stimulate better understanding by prompting the user to answer questions about typical attributes of a situation, providing information that is prototypical for known classes of situations and revealing whether solutions or partial solutions are available. An understanding aid can prompt the user to make the understanding of goals explicit. Aids of this type have risks associated with over generalisation; conditions characteristic of some class of situations are confused as some other class. Considerable domain knowledge may be necessary for this type of strategy.

514. Choice. The comparison of options or the structuring of human judgement with quantitative methods is a common approach to aiding. The belief is that through systematic and numerical means to judgement, choice can be improved by making it more objective. Decision analysis techniques like multi-attribute utility (MAU) and cost-benefit-risk calculations are typical of this approach. Many of these approaches are compensatory where the values of different attributes are placed on compatible scales suitable for trade-offs. Noncompensatory approaches are those in which trade-offs among attributes are not allowed. Comparisons are made on an attribute-by-attribute basis relative to a standard or based on comparisons among options. Payne, Bettman, and Johnson (1988) provide some indication when noncompensatory comparisons can be more accurate than compensatory ones.

515. The following noncompensatory strategies for selection are adapted from explanations by Hwang and Yoon (1981):

a) A dominance strategy is one where an alternative is dominated if there is another alternative which excels it in one or more attributes and equals it in the remainder. The number of alternatives can be reduced by eliminating the dominated ones.
b) A "maximin" strategy is one based on the saying that a chain is only as strong as its weakest link. Overall performance is determined by the weakest or poorest attribute. A "maximax" strategy is based on the greatest value is placed on the best attribute value rather than the poorest one as in maximin. "Maximax" is similar to disjunctive methods where alternatives are evaluated on the greatest value of any attribute; or where a premium is placed on any extreme value (a "stand-out" quality).
c) Satisficing is based on the concept that an option must meet or exceed minimal values or standards to be acceptable.
d) A lexicographic strategy is where a selection is made on the most important attribute, unless several options are tied, then the next most important attribute is used to compare options and so on.
e) Elimination by aspects is a procedure in which all options are removed from consideration if they do not satisfy some standard level, continuing until all alternatives except one have been eliminated.

516. **Planning.** In addition to reasoning for selection, reasoning done to construct or design options or plans is also important. Several subdisciplines of operations research provide techniques for deriving optimal solutions (also known as "ideal" and "nondominated" solutions). Linear programming, goal programming, and dynamic programming are some of the mathematical approaches to minimise and/or maximise goal and constraint variables. Planning can also be assisted with domain knowledge and procedures which attempts to meet goals and constraints by qualitative means. Aids have been developed that are relatively successful in evaluating plans. Another approach is to aid the detailed assignment, allocation, and scheduling of actions. A synchronisation matrix is one approach for doing this. A related strategy is to assist with the development of contingency plans. Contingency plans are developed to prepare for multiple future events. Contingency plans "hedge" against uncertain future actions and outcomes.

517. **Prediction.** Another approach is to predict what might happen in the future or what is likely to happen in the future. Again there is a useful dichotomy of qualitative and quantitative techniques for prediction. Qualitative predictions are based on conceptual rules, while quantitative techniques either apply standard statistical approaches or quantify judgements. Regression and time series models are example approaches for quantitative prediction.

518. **Expertise.** Expert system strategies make up a broad category in this taxonomy. Further definition of expert systems requires knowing what expert performance is for specific applications. (The section (6.3.4) on knowledge elicitation provides information about different types of knowledge structures and how to derive them.) One type of expert systems is experience- or case-based strategies that try to encode conditions and responses from previous events for application to future use with the expert system. Another subcategory of expert systems assists with heuristic-based judgements. Heuristics like representativeness and availability that might be used by experts can be embedded in an aid. The heuristics are generally based on approximations, useful when complete information is unavailable. The characteristics that make these useful heuristics in some cases can also be liabilities in other situations when the approximations are not adequate.

6.5.2.3. Information Handling

519. This category focuses on strategies which address the manipulation of information. The strategies in this category rely on the computational strength of computers. They allow the computer to assist with the functions that it excels, to unburden the decision maker of routine tasks, or to create unique representations. The concepts include a mixture of general-purpose tools (like spreadsheets, data bases, word processors) and displays of information. Information handling tools may be easier to develop and implement, but they may impact decision making performance the least. The precision and completeness that computers demand may cause an overall increase in workload.

520. **Patterns.** Spatial orientations are important for recognition of patterns of terrain, enemy

formations, and other battlefield features. Analysis of these patterns is important for understanding critical terrain features, sensor coverage, movement envelopes, and predicting possible locations and activities.

521. **Graphics.** Graphical displays can provide powerful representations of information to commanders and other personnel who need information that can be assimilated quickly. Bar charts, histograms, trend lines, relational diagrams, and many more techniques can be used to present integrated information.

522. **Maps.** Geographical and topological displays can be used to display features and calculated results of associated with various map features (like elevation bands or effects of rainfall on mobility). Traditionally, paper maps, acetate overlays, and manually-generated status displays were the primary modes of graphical displays. Operators can now have the same data quickly displayed on the computer screen, with the capability to rapidly mix and modify the information for optimal viewing and exploration of relationships. Aids that perform calculations on graphical information can provide further assistance in dealing with time-distance relationships.

523. **Uncertainty.** Uncertainty is a pervasive characteristic of complex decision problems. Strategies which deal with uncertainty try to reduce the uncertainty or find ways of accounting for it. Some procedures involve improved weighting of uncertain or unreliable data, other procedures try to account for a fuller range of possibilities by estimating the intervals between highest and lowest values rather than specific point estimates.

524. **Accuracy.** Spreadsheets can be used to take the computational burden off the human. They also can serve to provide a schematic representation of problems and solution states. The advantage of computer calculation over human abilities is well-documented. The risks associated with spreadsheets is imposing too much mathematical structure on problems, that might exceed thresholds of human input or challenge understanding.

525. **Statistics.** Summarising or interpreting numerical data is not usually considered a decision aiding strategy, but certainly is a valid concern when dealing with quantitative information. Determining central tendencies, variability, and comparing observations to standard probability distributions can prove useful.

526. **Memory.** A universal decision aiding strategy is to implement the concept of extending human memory. Both storage and retrieval can be operated on. Typically this strategy augments human memory with computer storage to provide another means to information that can be stored, retrieved, and operated on by predetermined procedures. Data bases and knowledge bases offer the opportunity to increase memory, but has the danger of relegating the user from a solver of problems to a handler of information. The focus on solving problems can subtly shift to data base operator. Simple means of extending memory allow the user to store notes in the computer and to organise them in a way useful to the task.

527. **Documents.** Automatic generation of text, text editors, and word processors and similar office productivity tools can reduce workload and allow more complete record keeping. Many of these tools exist in commercial form. These multi-purpose tools can be applied to support the management, calculation, production, and communication of information.

6.5.3. Guidelines for Designing Decision Aids

528. There are remarkably few sources of decision aiding guidelines that have been compiled. This is in contrast to human computer interface design where there are numerous sources of guidelines (see Section 5.2.2.1). Although easy-to-use guideline documents are not available, separate guidance does exist on various aiding issues. The guidelines are dispersed throughout the literatures from different disciplines like psychology, human computer interaction, decision sciences, operations research, management, computer processing, and economics. Some selected sources addressing decision aid guidelines include (Andriole, 1989; Carroll & McKendree, 1987; Jacob, Gaultney & Salvendy, 1986; McCann, 1989; Murphy & Mitchell, 1986; Ramsey & Atwood, 1979; Rook, 1984; Sage & Rouse, 1986; Samet, 1984; Tolcott & Holt, 1987). If COADE and the cited references do not provide

guidance on specific topics of interest, chances are that relevant information exists elsewhere.

6.5.3.1. Guidelines Derived from Failed Decision Aids

529. Many decision aid attempts have not realised the success that was desired. In most cases of unsatisfactory performance there has not been clear diagnosis and reporting of the failures. The reluctance to report on development failures is understandable, but contribute to a cycle of repeating flaws. The following guidelines are based on generalisations of direct experience and open sources that corroborate these observations. The guidelines provide cautions about common traps that development efforts have fallen into.

530. Requirements. **Base design requirements on analysis.** Often decision aiding efforts begin with the notion that a technology or approach could be used for some job or task, instead of examining the task first for performance deficiencies or opportunities. The actual ways decisions are made are ignored, when a technology (like hypertext) or specific technical approach (like a normative choice method) is assumed to be desirable. The solution approach is selected well before the nature of the task, the decision makers, and the cognitive processes are sufficiently understood. Often the development or application of some technology becomes the goal, rather than improving decision making performance. Solutions cannot be force-fit, using the latest technology available or a favourite approach that a designer (or analyst) believes to be rational or normative (McCann, 1989).

531. Functionality. **Determining what the aid does needs to be a deliberate, justifiable process.** The selection and design of aiding strategies are complex activities. What the aid does and how it represents the context in which it helps are perhaps the most difficult aspects of effective aids. Aid proponents sometimes view aiding as automating parts of a task, providing support for what the users request, or designing enhanced information displays. While any of these approaches might lead to a workable aid, ignoring the functionality makes any candidate performance improvement only a gamble. The function or behaviour of an aid must be considered before the interface issues. What the aid intends to do, its goal, and strategy for doing it must be dealt with, even at the cost of neglecting "look and feel" issues (Fallesen, 1991).

532. Scope. **Develop aids that focus on specific problems.** Several decision aiding efforts have been quite ambitious, hoping to automate all aspects of mission planning or information processing. This type of approach is based on the notion that information once stored in a computer form can be used to support multiple tasks. And once the process and results of tasks are represented, then that information can be used in subsequent tasks. Such exhaustive representation and storage may be useful if tasks are straightforward and highly proceduralised. C2 tasks do not usually have these characteristics. C2 involves uncertain information, goals, and procedures. The associated thinking is not easily codified in computerised form. Generally, the "grand" aid that promises to do everything does not work well (Fallesen, 1991; Nickerson & Feeher, 1975). The grand approach tends to focus too much on computer representation instead of performance.

533. Experience. **Avoid repeating decision aiding failures.** Decision aiding does not seem to be easy to implement for military applications. There have been numerous attempts at trying to develop decision aids for all kinds of applications. These previous efforts can provide valuable information when considering various approaches. Many aiding strategies have been ill-conceived and not particularly powerful to use in operational settings. After some aids have been developed, they are found to be fairly trivial, containing a few good rules that could have been more easily and better included in training or non-aided procedures. Analysts and designers need to identify and study previous attempts at aiding in predecessor or reference systems. These experiences should provide valuable information for why particular strategies failed or succeeded.

534. User acceptance. **Make explicit trade-offs between performance and user acceptance; do not simply defer to users' opinions or preferences.** User desires should be assessed and taken into account, but users should not set requirements themselves. Users' opinions should not be the only source of assessment of concepts or prototypes. Users do not necessarily have a good understanding of their performance. It is very difficult for proficient decision makers to describe how tasks are done since their abilities can be automatic. Users can be so set in the way that a task is routinely done that their suggestions may be narrowly constrained. Users may not know about alternate approaches or technologies. What users can envision may not be the same as what will improve their performance.

If users have difficulty accepting effective aids, then the organisation should find ways of encouraging use (see Riedel, 1988 for user acceptance guidelines).

535. Flexibility. **Provide options to the user.** Build flexibility into the aid. Provide a resource "envelope," a tool box (McCann, 1989). Aids should not place barriers in procedures or concepts where they did not exist in an unaided environment (Fallesen, 1991; Weber & Coskunoglu, 1990). An aid should not be more constraining than manual procedures, unless that is the intent of the aid because some specific procedure has been found to improve performance. A desirable trait of aids should be that they are robust to accommodate a wide range of situations or to degrade gracefully over progressively more complex situations.

536. Accuracy. **Do not invent relationships among information.** Deterministic relationships among information are often developed to allow the use of algorithms, provide compatible comparisons, or maintain similar data structures. In one aid, an algorithm was developed for estimating resource consumption. One algorithm computed consumption based on a ratio of personnel in a subordinate unit to its parent unit, rather than using the amount of actual equipment in that unit and its projected missions (Flanagan, 1993). Although invented relationships may not be incorrect, they at least can be unfamiliar and misleading, especially in novel situations. Proposed relationships should be verified before they are incorporated into an aid. The implications of those computations need to be examined in light of other computations in the system and for future situations. Once an algorithm is embedded in an aid, it can be difficult to isolate it, trace its origins, and to correct any interdependencies that have been established with other algorithms.

537. Precision. **Do not impose a computer determined level of precision and formality on decision makers.** Computers' strong points are that they allow perfect replication and storage and error-free computations. The more that a task can be framed using exact data and rules, the more straightforward it will be for the computer. Rigidity and exactness are traits that are not strong points of human decision makers. Humans' abstraction and induction abilities are powerful for coping with complex and unfamiliar problems. Avoid false precision in aids. Do not use excessive significant digits and single measures of central tendencies. Alternatives that are more natural for humans and realistic for complex problems include fuzzy logic, confidence intervals, upper and lower bounds, comparison of worst and best possible outcomes, and sensitivity analyses (Fallesen, 1991).

538. Training. **Identify training requirements incurred because of the aid.** Aid proponents sometimes make the argument that their aid will reduce training and personnel. Decision makers generally need to retain at least an equal understanding of what to do in the task (even if the purpose of the aid is to inform). In fact, there is likely to be a net increase in training requirements. Besides the usual task knowledge, decision makers need to monitor the accuracy of the aid, understand the output of the aid, and know the boundary conditions of what does and does not apply. Decision makers should retain the responsibility for the task and will be the ultimate back-up if the aid cannot be used. An aiding concept that promises to obviate the need for training or reduce training costs, personnel qualifications, or manpower may be 'too good to be true.'

539. Context. **Anticipate the context into which an aid will be introduced.** An operational concept needs to be developed for the environment in which the decision aid will be used. This is especially important if the decision aid is mostly driven from a technological opportunity, rather than an analysis of decision aiding needs. An aid is supposed to create change, but some of the changes may be unexpected or undesirable. An analysis of system changes should be done concurrently with analyses and design and all phases of evaluation.

This page has been left blank intentionally

6.6. HUMAN-COMPUTER INTERFACE DESIGN

6.6.1. The Concept of Usability
6.6.2. Selected Design Principles, Guidelines, and Standards
6.6.3. Design Methodologies and Tools
  6.6.3.1. Design Methodology
  6.6.3.2. User Interface Design Tools
  6.6.3.3. Rapid Prototyping
6.6.4. Models and Architectures
  6.6.4.1. The Use of Models
  6.6.4.2. Dialogue and Interaction Architecture
  6.6.4.3. Information Exchange
6.6.5. Promising Interaction Techniques and Technology
  6.6.5.1. Intelligent Interfaces
  6.6.5.2. Multimodal and Multimedial Dialogue Design
6.6.6. Dialogue Summary

540. The COADE framework defines a decision aid as "a computer-based system extending the human cognitive capabilities to adapt to the requirements of the task to be performed". According to this definition, the user of a decision aid must cooperate, (i.e. interact and communicate) with a computer ; the computer is seen as a partner or tool. The human-computer interface (HCI) provides the means by which the collaboration and the communication about intentions and doings of the two partners occurs — that is, the human-computer dialogue. Modern computer-based systems attempt to provide sophisticated human-computer interfaces that make use of the most recent results of information processing and information representation theory and technology. The HCI of a computer-based system concerns itself with the computer, the user of that computer and the way in which they interact. It is composed of those parts of the machine that are apparent to and manipulable by the user; and those models and impressions that are built up in the mind of the user in response to interacting with these features. Thus the HCI incorporates elements that are part of the computer, elements that are part of the user, and methods for communicating information from one to the other (Barfield, 1993). However, it is not sufficient to consider the human factors (perceptive and motoric aspects) of the physical interface. Of equal or greater importance is to address the cognitive aspects of Command and Control (C2) tasks, the exchange of information between user and decision aid, and to account for the characteristics of the user, i.e. amplify strengths and compensate for weaknesses (Hollnagel, 1991) by developing intelligent and adaptive interface designs.

541. The current generation of C2 systems provides strategic and tactical decision makers with an unprecedented quantity of information. The way in which these systems present this information and enable the dialogue between decision maker and the system is often identified as a limitation of their overall effectiveness. While there is widespread interest in making the user interfaces to C2 systems more usable, and less costly and time consuming to build, a consensus on the best ways to achieve these objectives has yet to emerge (Andriole, & Halpin, 1986). There is little doubt that the efforts and costs associated with developing HCIs are substantial. Software for the user interfaces has been estimated to represent from 30% to 50% of the software making up typical interactive computer systems and interfaces are frequently described as one of the most difficult and time consuming components to develop (Myers & Rosson, 1992).

542. There is currently no widely accepted unified theory or model of human-computer interaction or of the process of designing the interface against which to judge the value of existing systems or to define requirements (Hollnagel, 1991). There is also relatively little consensus concerning criteria on what constitutes a good interface design. About the only thing on which most human factors experts will agree is that to produce a good design, the specific, intended purpose of

the system must be known. Unfortunately, this implies that the interface design problem will be difficult to solve in general. Every new interface design problem will probably require the acquisition of substantial knowledge of the problem that the system is intended to support, i.e. knowledge about the user, the task, and the environment.

543. Currently, two main approaches exist for improving the human-computer interface component of a system so that overall system performance is improved. They are display enhancement and intelligent interface design. These two approaches stem from the two principle ways for aiding human performance:

a) people can be aided in what they perceive (by making important information more easily identified); this is the chief premise of display enhancement;
b) people can be aided in what they do with the information they perceive (making it easier to perform operations, etc.), which is a goal of intelligent interfaces.

544. From the literature it appears as if novices, who work more with context-free elements and rules and are not able to identify subtle differences, can be said to be working on a level which is best described using rules. If this is the case, an intelligent interface may make more sense to them because it appears to be making decisions in a way similar to the processes that the novices themselves use. In contrast, experts behave more intuitively and are very context dependent. Therefore, enhanced displays may be more consistent with how they view the problem domain (Kirlik, Markert & Kossack, 1992).

545. Because of the range of problems that computers can be used to solve, and the rapid development of new interface technologies, interface design has become a challenge that is more difficult than ever before. Design of HCIs, like most areas of design, is a process that is difficult to describe in terms that are both precise enough to be useful to people doing specific work, and general enough to remain relevant for any reasonable period of time, or for more than one specific application. Therefore this section can give only a snapshot for design activities and the relevant research and development.

6.6.1. The Concept of Usability

546. The necessity of a user, task, and situation oriented HCI design approach, i.e. designing with respect to the abilities and skills of the potential user, the specific tasks to be performed, and the situational and environmental factors shaping task performance has been argued in COADE. The philosophy of focusing on user requirements instead of on technical and technological innovations (Norman & Draper, 1986) is widely accepted. In this section, the concept of usability will be outlined. Usability must be distinguished from utility (or usefulness), which is fitness for purpose, i.e. how useful a system or software product is. This is quite distinct from usability. A product can be very useful but difficult to use. Thus, utility is related to purpose whilst usability is related to means. Utility essentially covers the functional capability of a product. It is measured through indicators like capacity, speed, reliability, flexibility, accessibility and through cost or economic indicators (Alty, 1992).

547. From an ergonomics standpoint, a well-designed man-machine system means that the technical component is usable. A definition of usability was first put forward by Miller (1971) in terms of "ease of use". The concept was further developed by Bennett (1979). It is now generally agreed that usability depends upon the dynamic interplay of four components (Shackel, 1984):

a) the user
b) the task
c) the environment
d) the tool.

548. Good user-computer dialogue design therefore requires an understanding of all four components and the dynamics and complexity of interaction between them. Criteria associated with the aspect of usability are (Bennett, 1984):

a) learnability ease of use, training time and relearning time for intended users
b) effectiveness system's capability to provide problem solutions of the desired

quality in an acceptable manner (speed, error, warm-up time, etc.)

c) flexibility system adaptation to a variation in tasks and/or changing work environments

d) attitude user opinions about continuous use (fatigue, discomfort, frustration, and personal effort)

Thus, the techniques associated with usability engineering are concerned with observing users in their work context, with setting usability goals, and with the attainment of those goals. To this end, integration of user participation in the design process may be beneficial (Kallela, 1992).

6.6.2. Selected Design Principles, Guidelines, and Standards

549. The design of user interfaces requires a considerable degree of domain-specific knowledge and problem solving skill. Interface principles (from research and experience), guidelines (from applied research), standards (established by authoritative agencies and industry), and rules (tailored guidelines) (Deimel, 1988) are the traditional medium for transferring specialists' expertise to designers. They describe the conventions and practices for designing user interfaces that promote consistency, ease of learning, and ease of use of the resulting computer-based product. They address such topics as the design of dialogues between users and computers, information representation, control devices, and effective error and help procedures.

550. Though the human factors and the software ergonomics literature abounds with guidelines and recommendations for HCI design, most of them are written at a level that presumes they will be applied by professionals with considerable expertise , background and relevant "common sense". This makes them appear very abstract. It is common for information on interface design to be published in a fragmented form rather than compiled in a comprehensive interface design prescription or methodology. Guidance is typically presented in terms of (Papazian et al., 1989):

a) a general philosophy, principals or goals (i.e., Norman, 1986);

b) a prescriptive model for the design process (i.e., Williges, 1987);

c) guidelines and criteria for assessing the quality of good designs (i.e., Smith & Mosier, 1986);

d) national and international or industrial standards for improving usability and consistency (Thovtrop & Nielsen, 1991).

551. If guidelines are to be useful, they must present specific, relevant design rules. Platitudes and vague advises are of little value to a designer. A good test of a guideline's quality is the extent to which it serves as audit criteria: if designers cannot readily determine whether an existing HCI design meets a guideline, the guideline probably needs to be clarified. Guidelines also need to be realistic and to reflect the hardware and software constraints of the system to be designed. If general guidelines are intended to be applied in a specific setting with known constraints, they should be tailored to those constraints (Brown, 1991). Because the establishment of user interface standards is based on the widely held feeling that consistency is one of the most important usability considerations, it seems reasonable to demand usability in the standards themselves (Thovtrop & Nielsen, 1991).

6.6.3. Design Methodologies and Tools

6.6.3.1. Design Methodology

552. On the assumption that the quality of a system is influenced by its development process, the use of the right development methodology will be critical. A methodology draws together theories, models, principles, approaches, and processes and organises them to contribute in concert to produce a usable and functional product (Hartson & Boehm-Davis, 1993). A design methodology specifies what decisions are to be made, how to make them, and in what order.

553. As previously mentioned, there is currently no accepted unified theory or model of the HCI design process, although there exist some methodologies to be followed during design. Design can be viewed as an information transformation process that includes several phases. The number and designation of these phases, as well as their definition and relationship to each other differs among

researchers (Rouse & Boff, 1987a; Sage, 1991; Watermann, 1986). Most methodologies include a problem identification or problem formulation phase; a conceptualisation or solution finding phase; an implementation phase; and a testing and evaluation phase. Despite the differences, there seems to be general agreement that the design process should be evolutionary or incremental (Buchanan, Barstow, Betchtel, Bennett, Clancey, Kulikowski, et al., 1983) and that because it is an iterative activity, it will involve some parallelism (Rouse, 1991). The design process is embedded between the corresponding development phases of analysis and summative evaluation.

554. The design of the HCI for complex systems benefits from the involvement of people with skill and experience in different areas. HCI design experts are typically human factors engineers, cognitive psychologists, or software engineers who specialise in the field of cognitive ergonomics with special interest in usability aspects. In addition, the participation of end users, who have the benefit of task experience, should be mandatory to ensure integration of the design and evaluation phases (Kallela, 1992).

6.6.3.2. User Interface Design Tools

555. User interface programming is a time consuming and expensive task in today's human-computer system development process. Therefore, there is great interest in the development of tools to help in the design and implementation of interfaces. Software tools for user-computer interface design or development — often called user-interface management systems (UIMS) or user-interface development environments — are themselves interactive systems that support production and execution of the HCI. They include user interface development tools, toolkits, and development environments.

556. Interface development tools range from simple interface routines to complete interface development environments. A toolkit is a library of routines that can be called to implement low-level interface features, and particularly, to permit the incorporation into the interface of various interaction techniques, e.g. menus, scroll bars, buttons, and their associated physical devices, such as mice, joysticks, trackballs. Toolkits are typically used by programmers who write source code to call the desired routines. They provide little or no support for an interface designer who is not a programmer.

557. A UIMS is an integrated interactive environment, often based on a broad set of tools, for designing, prototyping, executing, evaluating, modifying, and maintaining interfaces. In particular, a UIMS supports the complete development and life cycle maintenance of a user interface by a developer who may not be a programmer. This is commonly done through the use of graphical representation techniques for designing and prototyping an interface, as well as run-time execution support so that it can be evaluated and iteratively refined. Norman and Draper (1986) summarise the use of a UIMS: A UIMS provides a way for a designer to specify the interface in a high-level language. The UIMS translates that specification into a working interface, managing both the details of the display and its associated input and output and also its interaction with the rest of the program. By relieving a developer of much of the burden of coding and programming in producing an interface, he can concentrate on the design itself, rather than on its implementation (Hix & Schulman, 1991).

6.6.3.3. Rapid Prototyping

558. Rapid prototyping, user testing, and redesign are key components of the iterative HCI design process. The development of prototypes permits original design concepts and redesigns to be turned into something concrete. Although rapid prototyping is primarily a technique, not a tool, it calls for dialogue development tools that permit quick production of prototypes to allow early observation of interface behaviour as well as easy modification of designs. With rapid prototyping, the design process is accelerated so that alternatives can be evaluated and the effects of each modification observed (Hartson & Hix, 1989).

Rapid prototyping is an effective way to involve the user in system design. The prototypes make design more understandable to the users, force the designer to work out the details, and serve as a concrete proof of abstract design concepts. The testing of prototypes provides an active user role in system development, user commitment to the system, and user acceptance of the system (Brown, 1991). Prototypes will allow designers and users to experiment with some or all aspects of the interface function, including its "look and feel".

559. Different levels of prototype detail, functionality, and fidelity are used in the design-redesign cycle. At the lowest level of prototype fidelity is the graphic "storyboard" approach where only the superficial look of the presentations is depicted; storyboards have been shown to be useful for eliciting user impressions. Interactive dialogue presentations that simulate both the look and the feel of the interface offer the next level of realism, followed by facades that incorporate simulated system responses and response times. The highest fidelity and most useful prototype is one that includes the running application software and additional data collection routines for quantitative user-system performance evaluations.

560. Each of these prototyping methods has strengths and weaknesses that make it difficult to rank them. The storyboard approach is least costly in terms of time and skill required. Facades take more time to produce, but for evaluation purposes provide the dimension of feel that storyboard do not. Finally the fully functional prototype provides the most realism but at the highest cost in terms of time, programming, and skill requirements. In many instances, the uncertainty associated with initial designs do not warrant the investment in completely operational prototypes. Referring to Beevis and St Denis (1992), it appears that rapid prototyping facilitates an iterative approach to the design of the HCI, and that it is most applicable to the early stages of system design, rather than to detailed design.

6.6.4. Models and Architectures

561. Models and architectures are a somewhat abstract and general approach for describing problems in a structured manner, one that focuses on the essential aspects for problem solving. The establishment, adaptation, and use of models or architectures leads to better understanding of a problem, stimulating and improving the solution process.

6.6.4.1. The Use of Models

562. The idea of building and using models as a basis for design of interfaces for cognitive support of user's tasks is of recent origin. Cognitive model-based design approaches have been constructed and applied e.g. by Rouse (1981), Rasmussen (1986), Woods and Hollnagel (1986), and Zachary (1986, 1988). The idea is to develop an interface design compatible with the user's model of the system as well as with his task procedures, and by doing this to enhance the efficiency of the overall system performance. User and task modelling is an important aspect of the design of systems that seek to adapt their behaviour to users, permitting more flexible and intelligent interaction (Kass & Finn, 1991). There exists, however, some confusing terminology with reference to user models. Depending on the context and the author, a "user model" may be one of the following:

a) the users' model of the system they are working with
b) the representation of the users as embodied in the system
c) the model of a user as applied by others, e.g. the system designer
d) the designers' model of users' model of the system
e) the users' knowledge (and beliefs) about themselves

563. To make the confusion even greater, a number of different terms, including user model, mental model, and conceptual model, are applied in reference to one or more of these five meanings (Hollnagel, 1991; Zachary & Ross, 1991). Although user models have been incorporated in many types of interactive systems, there are no general unified user models that encompass a range of users, tasks, and systems. These models must be specifically crafted for each application, usually by explicit coding of domain-related goals, plans, or knowledge that users are expected to have (Kass & Finn, 1991). Models are good predictive tools, but have not yet proven their utility in the time- and cost-constraint system development cycle (St. Denis, 1990)

6.6.4.2. Dialogue and Interaction Architecture

564. Modular architectures for user-computer interaction were developed with the goal of structuring the complex interaction between humans and computers when the computer is treated as a tool to carry out a task (Hollnagel, Mancini & Woods, 1986). Recent developments in human-computer interaction recommend a clear separation between application and user interface (Hardy & Klein, 1991). The attempt to separate user interface and application functionality of the computer and

to develop the resulting parts more or less independently has resulted in new conceptual and architectural models that describe the interaction aspects of a system in more detail. The SEEHEIM model (Green, 1985) was elaborated during the IFIP workshop 1983. It was the first model for user-computer interaction with three modules: Presentation, Dialogue Control and Application Interface Model.

565. The IFIP model (Dzida, 1983) which has a more detailed architecture uses three independent interfaces to separate the user of a human-computer component from the application layer:

a) presentation or input and output interface,
b) dialogue interface,
c) application interface

A fourth interface is the organisation interface which builds the bridge to the work environment. The idea of modularisation and functional separation of the different aspects of user-computer interaction is very useful. There is a clear structure for the communication process and each of the dialogue modules can be developed, redefined, evaluated, and maintained separately and in parallel by specialists in the field.

566. Different user-interface management systems take different philosophies in managing the user-computer dialogue. The management system controls the dialogue between user and system through the user-system interaction and communication sequences. Its major function is to take input from the user and give output from the computer. It calls different action routines in response to user actions to, for instance, modify the dialogue by restricting what the user can do next. Some user-interface management systems embody the philosophy that the control resides strictly in the management system thus giving maximum isolation between user and application. Increasingly, however, the trend is toward shared control to allow the user interface to be more "intelligent" in guiding the user (Marcus & Van Dam, 1991).

6.6.4.3. Information Exchange

567. Human-computer dialogue involves the exchange of information between two collaborating partners, the human and the computer. The problem of human-computer interaction can be viewed as two powerful information processors communicating with each other via a highly constrained user interface, using a traditional interface input and output technology, like keyboards or graphical displays with a mouse device. Therefore, the principle issue is how to present the relevant information needed by the partner at any point. Most of the research work done in the field of information representation at the human-computer interface focuses on psychological factors related to human perception, i.e. detectability and legibility of symbols, and relatively simple cognitive processes like short-term memory processes. But in reality, the effectiveness of information representation is only marginally influenced by factors concerning human perception. Characteristics of the environment and situation, the task, and cognitive capabilities of the user himself have a greater influence on the information exchange process. Furthermore, the increasing complexity of computers has resulted in an explosion of data at the interface. As Woods and Roth (1988) noted, the problem in complex systems is too much data and too little information. The problem is one of meaning and understanding: good "advice" is that which is given in situations which call for it, and "only when needed".

568. Woods (1991) makes an important distinction between interface design based on "data availability" and design for "information extraction". Availability of data or information alone can no longer be seen as an adequate reason for presenting it. Instead, optimum utilisation and presentation of available information should sought, otherwise the user is burdened with the difficult task of collecting, storing, and cognitively processing all potentially-relevant information.

569. Wickens (1984) states that the type of information structuring determines how and to what extent information is absorbed and stored. New information interacts with existing mental models of situation and events. Information representations that are compatible with existing mental models improve absorption and storage.

570. The interface to a decision aid should be designed for the task of information aggregation,

i.e., collection, correlation, integration, and fusion, as well as for the selection of quantity and quality, i.e., type, form, and organisation or structure of the information to be displayed.

6.6.5. Promising Interaction Techniques and Technology

6.6.5.1. Intelligent Interfaces

571. One way in which user-computer interaction could be improved, quantitatively and qualitatively, is by relieving the human user of the burden of adapting to the system by having the system adapt to the user i.e., the system has some degree of flexibility (Hollnagel, 1991). Intelligent interfaces are designed to automatically adapt the representation of the information to the context of the user; they act like an intelligent assistant, available to support the user if necessary. Issues in the design of intelligent interfaces include such topics as adaptive user guidance for task accomplishment, task and user-adaptive control of dialogue, task and user-adaptive information representation, and effective error and help procedures. However, the problems that are implicit in adaptive interfaces are fundamentally difficult and complex (Norcio & Stanley, 1989).

572. One of the first problems that must be addressed is the issue of user models. Models must necessarily be based on theories of cognition and must explain differences in user capabilities and evolving changes in user performance. They must be able to deduce user levels of expertise and experience by collecting input parameters such as command types, error rates, and speed. Another critical aspect of an adaptive interface is the dialogue between the user and the system which must be appropriate for the specific user as well as for the specific task. A third important issue is the structure and architecture of the interface which must be an integral part of the overall system so that the adaptation can take place in the context of the application.

573. Although the idea of adaptive interfaces seems to have advantages, the concept does have its critics. First, the design of such interfaces will entail inevitable increase in implementation complexities and costs. Second, researchers argue that the user may not be able to develop a coherent model of the system if the system is frequently changing (Greenberg & Witten, 1985). Another problem that may arise with adaptive systems is the loss of control or the feeling of loss of control.

6.6.5.2. Multimodal and Multimedial Dialogue Design

574. The bottleneck in improving the effectiveness of interactive systems increasingly lies not in task itself but in the communication of requests and results between the computer and its user. In future, the dialogue between user and computer-based decision aids will not be restricted to traditional interface input and output technology, such as keyboards or graphical displays with a mouse. Research and development are focusing on the application of artificial intelligence together with the development of human-computer interface technology that will integrate speech input and output, natural language text, graphics, and pointing gestures to provide more comprehensive interactive dialogue. But the effective application of novel interaction technologies requires more than technological advances; sophisticated connectors are needed to control and synchronise the different media and vary the presentation mechanism according to the specifications of the user-task model, among others. Dialogues must be modelled on the manner in which people naturally communicate — in coordinated multiple modalities. The objective is to simplify operator communication with sophisticated computer systems (Sullivan & Sherman, 1991).

575. Multimedia and hypermedia interfaces having combinations of text, graphics, animation, and sound promise to be effective interaction environments since they combine communication modalities: studies have shown that people remember more if they combine seeing, hearing, and doing. On the other hand, embedding new media in interfaces can cause sensory overload, resulting in confusion instead of clear communication. Thus, integrating several media for a presentation becomes a complex task. The challenge is to create a single new media and a comprehensive user interface out of the several parts (Grimes & Potel, 1991).

6.6.6. Dialogue Summary

The user of the decision aid normally never sees data structures, inference algorithms, optimisation programs, and all the other procedures that go on inside the computer; any real aiding that takes place seems to occur through the interface. Thus the HCI is one of the most important components of a decision aid because, for the user, the interface constitutes the system (Wilson & Rutherford, 1989). Therefore the interface that permits the dialogue between user and computer must be designed very carefully, taking full account of the potential use, the tasks to be performed, and the situation and environment of the tasks (Hollnagel, 1991). The activities and guidance provided in the COADE framework are intended to help the design team to do exactly this.

6.7. EVALUATION

6.7.1. Introduction
6.7.2. Framework for Decision Aid Evaluation
6.7.2.1. Evaluation Factors
6.7.2.2. Relationship between Evaluation and Other Activities
6.7.2.3. Issues in Decision Aid Evaluation

6.7.1. Introduction

576. One of the most crucial issues surrounding the design and development of decision aids relates to their evaluation (Adelman, 1991; Adelman & Donnell, 1986). Specifically, only through a careful and rigorous evaluation of decision aids as they are conceived, designed, and fielded will it be possible to: 1) ensure that effective systems are built, and 2) contribute to a general understanding of how decision aids should be built in future endeavours. Unfortunately, for a number of reasons, the matter of evaluating decision aids is a complex one, and one that is often overlooked in actual development efforts. Indeed, few documented efforts to determine decision aid effectiveness can be found.

577. The purpose of this section is to provide a framework in which decision aid evaluation can be conceptualised, and to describe some of the common pitfalls and challenges associated with actual decision aid evaluations. To accomplish this, we first present a framework for decision aid evaluation that incorporates several important evaluation concepts. Using this framework, we provide a summary of the evaluation activities we believe are necessary to assess fully a decision aid's effectiveness (Section 5.3. provides detail on these). Next, we provide a description of how we believe evaluation activities are related to other aspects of decision aid design and development, and describe our philosophy regarding decision aid evaluation. Finally, we describe several issues that pertain to decision aid evaluation. These are offered for the purpose of highlighting common problems in evaluating, and hopefully stimulating thinking about how to overcome them.

6.7.2. Framework for Decision Aid Evaluation

6.7.2.1. Evaluation Factors

578. There are literally countless way to approach and conduct the evaluation of a decision aid. So many, in fact, that developers are often overwhelmed by the task, and chose to ignore it. For example, the following questions are all related to decision aid evaluation, but are clearly concerned with vastly different aspects of the decision aid's functioning:

a) Will the user population actually use the final product?
b) Does the decision aid improve operational performance?
c) Is the knowledge base underlying the decision aid consistent?
d) Was the decision aid's software written efficiently?
e) Does the decision aid require additional training?
f) Are there side effects associated with the decision aid's introduction?
g) Will the decision aid's payoff justify its cost?
h) Was the cognitive task analysis conducted prior to development thoroughly done?

579. Any of the questions shown above (and many more) can reasonably be asked regarding a decision aid, and the way it functions. Therefore, the phrase "evaluating a decision aid" actually refers

to a family of related activities all concerned with gathering evidence in support of the decision aid's effectiveness. This means that decision aid evaluations can be conducted for different purposes, at different times, using different criteria and employing different methods.

580. In order to make the task of planning an evaluation more manageable, a number of defining factors can be considered. These factors are crucial in that they help to determine how the evaluation is designed, what is measured, how measurement will be accomplished, and so forth. They include: the purpose of an evaluation, the nature of criteria used in an evaluation, and the methods employed to accomplish an evaluation. Each of these factors is described in more detail below.

581. **Purpose of the Evaluation:** First, it must be determined why the evaluation is being conducted, for what purpose the information is being obtained, how the information will be used, and who will use it. While this seems like an obvious point, it is often overlooked; that is, practitioners or system developers feel compelled to conduct an evaluation, but are not sure why, or how the information will be put to best use. Consequently, time and effort are invested in an evaluation that does not yield meaningful information. Moreover, the purpose of an evaluation drives the questions that are asked, what data are collected and how data are gathered.

582. **Nature of Criteria:** Second, it must be determined what exactly is being measured, which data are most useful to answer the questions of interest, and how performance (and, more specifically, effectiveness) is defined in the task domain. There exist literally hundreds of potential criteria that may be appropriate for evaluating a decision aid. In order to focus the evaluation, the selection and/or development of criteria must be a function of the evaluation's purpose (i.e., the questions being asked), and not a matter or expediency or convenience.

583. **Selection of Methods:** Third, it must be determined how the evaluation will be conducted. Evaluation techniques and methods appropriate for assessing decision aid effectiveness have been developed in a variety of disciplines (e.g., human factors, psychology and other social sciences, cognitive science, education, and computer science). The selection of methods to accomplish a particular evaluation will depend on the two factors listed above, as well as other constraints (e.g., funds, time, availability of domain experts, etc.).

584. The factors listed above are useful in planning and designing an evaluation because they drive the answers to the why, what and how questions raised above. They can be used to help a decision aid developer in conceptualising an evaluation, and in ensuring that sufficient evaluation data are collected. Furthermore, it is our contention that the specification of these factors in any evaluation endeavour also depends in large part on when in the development cycle (i.e., the analysis, design or evaluation phase) of the decision aid the evaluation is being conducted. In fact, we would suggest that evaluation activities vary systematically as a function of the decision aid's phase of development. COADE Section 5.3 provides details regarding the types of evaluation activities associated with various phases of decision aid development. Table 6.3 summarises these activities, and indicates the purpose, methods and criteria associated with each.

6.7.2.2. Relationship between Evaluation and Other Activities

585. It should be noted that evaluation activities in COADE are seen as being iterative. That is, evaluation is conceptualised as being a continual part of all decision aid development activities--analysis, design and evaluation. In this sense, we advocate a "build a little, test a little" philosophy in decision aid development. There are numerous potential benefits to such a strategy. First, and most obviously, identifying design problems early in the development cycle can save time and resources from being wasted on a flawed approach. Second, early and continual user involvement can not only strengthen the decision aid's design, it can also help to ensure that users will "buy in" to the new system (and hence, use it). Third, the science of decision aiding will only advance when rich, detailed accounts of past efforts to incorporate cognitive concepts into decision aids are made available. The value of "lessons learned" in such a complex undertaking cannot be overstated.

Table 6.3. Summary of Evaluation Activities

Evaluation type	Purpose	Nature of criteria	Methods
ANALYSE			
Verify Behavioural Model	To verify the accuracy and completeness of the behavioural model	Task-specific; based on task and other analyses	Expert opinion; modelling; simulation
Verify Cognitive Model	To verify the accuracy and completeness of the cognitive model	Task-specific; based on cognitive analysis	Expert opinion; simulation; experiments; quasi-experiments
Validate Cognitive Requirements	To verify and/or establish the hypothesised cognitive requirements	Task-specific; based on cognitive analysis	Experiments, quasi-experiments, case studies, observation
Verify Performance Measures	To verify and refine performance measures, standards and criteria	Task-specific; based on thorough task analysis	Experiments; expert opinion
DESIGN			
Verify Design Concept	To assess the soundness of planned design and implementation	Analytical; based on expert judgement	Structured interviews; questionnaires
Assess Adequacy of Design	To assess the probable effectiveness of the decision aid design	Task-specific; based on analysis	Rapid Prototyping; experiments; expert opinion
EVALUATE			
Assess Task and Organisational Performance	To determine the efficacy of the aid with respect to task and overall organisational performance	Task-specific; based on analysis	Experiments, quasi-experiments; observations
Evaluate decision aid Usability	To determine whether users can employ the system effectively	Reactions; HCI standards; frequency of use	Observations, questionnaires, interviews, expert opinion
Evaluate Technical Performance	To assess the technical performance of decision aid the program	Software oriented concerns; professional standards	Expert opinion

6.7.2.3. Issues in Decision Aid Evaluation

586. **Diagnosticity.** As we have noted, every attempt should be made to collect evaluation data that are diagnostic in nature. That is, data that can feedback to improve either the decision aid development process, or the decision aid itself. This is crucial for several reasons. First of all, the expense incurred in conducting an evaluation will not be well invested if the reasons for a decision aid failure are not revealed. It is not informative to find that a decision aid does not improve performance, without also figuring out how it might be improved. Secondly, future decision aid development efforts can benefit from the experience of other only when developers collect diagnostic data and document their evaluation efforts.

587. There are several approaches to evaluation that can help ensure that resulting data are diagnostic. To begin with, the emphasis must be on process. That is to say, the goal of evaluation should be to collect information regarding the manner in which the decision maker uses the aid,

noting particularly when the decision aid fails. Obviously, outcome measures are also important, but they only tell part of the story. A second strategy is to conduct evaluations iteratively as we have suggested. This sort of approach is useful because it forces the developer to assess the decision aid development process itself. It also provides proximal feedback, so that problems can be addressed before development proceeds.

588. **The Criterion Problem.** It is well accepted that there is no single, universally applicable criterion against which to evaluate the decision aid's performance. In fact, it is often difficult to determine what the "right" decision is in an uncertain C2 environment. This can cause difficulty when attempting to determine objectively whether a decision aid is effective or ineffective. While there are no definitive solutions to this problem, there are several strategies that can help to alleviate it. First, evaluation planning can specify in advance the criteria that will be used during the development process. It is particularly important to seek agreement among stake holders (i.e., developers, users, sponsors) regarding what will be the basis upon which decision aid effectiveness will be determined.

589. A second strategy is to employ multiple criteria where possible so as to harness a preponderance of evidence that a decision aid is successful. This is most easily accomplished via careful and thorough task analysis. That is, attempts should be made to establish multiple bases upon which a decision aid can be evaluated. Moreover, multiple measurements (over time) can help provide evidence of a decision aid's effectiveness.

590. **Logistics.** Often, some of the more difficult problems encountered in conducting an evaluation involve logistical issues. These include: availability of users (to serve as subjects), scheduling, sensitivity of information collected, expense, time, resistance, and organisational support. While some of these issues cannot be avoided (i.e., they must be dealt with as they occur), extensive, early planning may help to ease others. Clearly, scheduling, availability, expense, and to some extent organisational support are problems that may be solved if addressed early.

Accinelli, S. R., Robinette, M. L., & Jacobs, J. A. (1985). *G3 Analysis* (Report CAORA/TR-13/85). Ft. Leavenworth, KS: Combined Arms Operations Research Activity.

Adelman, L. (1991). Experiments, quasi-experiments and case studies: A review of empirical methods for evaluating decision support systems. *IEEE Transactions on Systems, Man, and Cybernetics, 21*(2), 293-301.

Adelman, L. (1992). *Evaluating Decision Support and Expert Systems*. New York: Wiley.

Adelman, L., & Donnell, M. L. (1986). Evaluating decision support systems: A general framework and case study. In S. J. Andriole (Ed.), *Microcomputer Decision Support Systems: Design, Implementation, and Evaluation*. Wellesley, MA: QED Information Sciences.

Alba, J. W., & Hasher, L. (1983). Is memory schematic? *Psychological Bulletin, 93*, 203-231.

Altman, J. W. (1967). Classification of human error. In W. B. Askren (Ed.), *Proceedings of Symposium on Reliability of Human Performance in Work* (pp. 3-16). Dayton, OH: Armstrong Medical Research Laboratory.

Alty, J. L. (1992). Can we measure usability? *Proceedings of Advanced Information Systems Conference. The New Technologies in Today's Business Environment* (pp. 95-106). Oxford, UK: Learned Inf.

Amalberti, R., & Deblon, F. (1990). Cognitive modelling of fighter aircraft process control: A step toward an intelligent onboard assistant. *International Journal of Man-Machine Studies, 36*(5), 639-671.

Anderson, J. R. (1983). *The Architecture of Cognition*. Cambridge, MA: Harvard University Press.

Andriole, S. J. (1989). *Handbook of Decision Support Systems*. Blue Ridge Summit, PA: TAB Books.

Andriole, S. J. (1990). *Getting it right!: Information system design principles for the 90's*. Fairfax, VA: AFCEA International Press.

Andriole, S. J., & Halpin, S. M. (1986). Information technology for command and control: Introduction to Special Issue on Information Technology for Command and Control. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-16*(6), 762-765.

Andrus, G. R. (1988). *Methods of Acquiring Expert Knowledge on Media Selection for Interactive Video.* Doctoral Dissertation, Peabody College for Teachers of Vanderbilt University.

Atkinson, G. (1990). Practical experience using an automated knowledge acquisition tool. *Proceedings of Second Annual Conference of the International Association of Knowledge Engineers* (pp. 87-97).

Baddeley, A. D., & Hitch, G. H. (1974). Working memory. In G. Bower (Ed.), *Recent Advance in Learning and Motivation* (pp. 47-90). New York: Academic Press.

Badre, A. N. (1982). Selecting and reporting information structures for visual presentation. *IEEE Transactions on Systems, Man, and Cybernetics, 12*(4), 495-504.

Bainbridge, L. (1979). Verbal reports as evidence of the process operator's knowledge. *International Journal of Man-Machine Studies, 11*, 411-436.

Barfield, L. (1993). *The User Interface, Concepts, and Design*. Wokingham, England: Addison-Wesley.

Barnard, P., Wilson, M., & MacLean, A. (1986). The elicitation of system knowledge by picture probes. *Proceedings of CHI '86* (pp. 235-240). New York: Association for Computing Machinery.

Beach, L. R. (1990). *Image Theory: Decision Making in Personal and Organizational Contexts*. New York: Wiley & Sons.

Beevis, D. (1992). *Analysis techniques for man-machine system design* (Report AC/243(P8)TR/7). Brussels, Belgium: Defence Research Group, NATO HQ.

Beevis, D., Essens, P., & Mack, I. (1994). *The application of human factors engineering in CF CCIS projects* (Report 94-33). North York, Ontario: Defence and Civil Institute of Environmental Medicine.

Beevis, D., & St. Denis, G. (1992). Rapid prototyping and the human factors engineering process. *Applied Ergonomics, 23*(3), 155-160.

Belkin, N. J., & Brooks, H. M. (1988). Knowledge elicitation using discourse analysis. In B. Gaines, & J. Boose (Eds.), *Knowledge-Based Systems* (pp. 107-124). New York: Academic Press Ltd.

Bennett, J. L. (1979). The commercial impact of usability in interactive systems. In B. Shackel (Ed.), *Man Computer Communication, Infotech State-of-the-Art Report*. Maidenhead, UK: Infotech International.

Bennett, J. L. (1984). Managing to meet usability requirements: Establishing and meeting software development goals. In J. L. Bennett, D. Case, J. Sandelin, & M. Smith (Eds.), *Visual Display Terminals: Usability Issues and Health Concerns* (pp. 161-184). Englewood Cliffs, NJ: Prentice Hall.

Bierbaum, C. R., Fulford, L. A., & Hamilton, D. B. (1990). *Task analysis /Workload (TAWL) User's Guide 3.0* (Research Product 90-15 (AD A221 865)). Alexandria, VA: US Army Research Institute.

Billingsley, P. (1990a). The standards factor: Standards on the horizon. *ACM SIGCHI Bulletin, 22*(2), 10-12.

Billingsley, P. (1990b). The standards factor: Committee updates. *ACM SIGCHI Bulletin, 21*(4), 16-19.

Billingsley, P. (1993). The standards factor: Standards on the horizon. *ACM SIGCHI Bulletin, 25*(4), 15-18.

Boar, B. (1984). *Application Prototyping: A Requirements Definition Strategy for the 80's*. New York: Wiley & Sons.

Boehm, B. W. (1976). Software Engineering. *IEEE Transactions on Computers, C-25*, 1226-1241.

Boehm, B. W. (1988). The Spiral Model of software development and enhancement. *ACM SIGSOFT Software Engineering Notes, 11*(4), 14-24.

Boff, K. R., & Lincoln, J. E. (Eds.). (1988). *Engineering Data Compendium: Human Perception and Performance*. Dayton, OH: Armstrong Aerospace Medical Research Laboratories, Wright Patterson Airforce Base.

Booher, H. R. (1990). *MANPRINT: An Approach to Systems Integration*. New York: Van Nostrand Reinhold.

Borgman, C. L. (1986). The users mental model of an information retrieval system: An experiment on a prototype online catalog. *International Journal of Man-Machine Studies, 27*, 47-64.

Brewer, W. F., & Dupree, D. A. (1983). Use of plan schemata in the recall and recognition of goal-directed actions. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 9*, 117-129.

Brewer, W. F., & Treyens, J. C. (1981). Role of schemata in memory for places. *Cognitive Psychology, 13*, 207-230.

Brown, C. M. (1991). Methodology for human-computer interface design. In D. Alexander (Ed.), *Industrial Ergonomics: Case Studies*. Norwood, Georgia: Institute of Ind. Engineers, Industrial Engineering and Management Press.

Bruner, J. S., Goodnow, J. J., & Austin, G. A. (1956). *A Study of Thinking*. New York: Wiley & Sons.

Bryan, G. L., & Regan, J. J. (1972). Training system design. In H. P. V. Cott, & R. G. Kinkade (Eds.), *Human Engineering Guide to Equipment Design*. Washington, DC: US Government Printing Office.

Buchanan, B., Barstow, D., Betchtel, R., Bennett, J., Clancey, W., Kulikowski, C., Mitchell, T., & Waterman, D. A. (1983). Constructing an expert system. In F. Hayes-Roth, D. A. Waterman, & D. B. Lenat (Eds.), *Building Expert Systems*. Reading, MA: Addison-Wesley.

Byrne, R. (1983). Protocol analysis in problem solving. In J. St, B. T. Evans (Ed.), *Thinking and Reasoning: Protocol Approaches*. London: Routledge & Kegan Paul.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum.

Card, S. K., Moran, T. P., & Newell, A. (1986). The model human processor: An engineering model of human performance. In K. R. Boff, L. Kaufman, & J. P. Thomas (Eds.), *Handbook of Perception and Human Performance, Volume II. Cognitive Processes and Performance*. New York: Wiley & Sons.

Carrier, H. D., & Wallace, W. A. (1989). An epistemological view of decision-aid technology with emphasis on expert systems. *IEEE Transactions on Systems, Man, and Cybernetics, 19*(5), 1021-1029.

Carroll, J. M., & McKendree, J. (1987). Interface design issues for advice-giving expert systems. *Communications of the ACM, 30*, 14-31.

Carter, J. C. F., Archer, M. A., & Murray, A. E. (1988). *Description of selected Army staff functions: Targets for planning aids* (Research Note 88-62 (AD A197 449)). Alexandria, VA: US Army Research Institute.

Cavanaugh, J. C. (1988). The place of awareness in memory development across adulthood. In L. W. Poon, D. C. Rubin, & B. A. Wilson (Eds.), *Everyday Cognition in Adulthood and Later Life*. Cambridge: Cambridge University Press.

Charness, N., & Campbell, J. I. D. (1988). Acquiring skill at mental calculation in adulthood: A task decomposition. *Journal of Experimental Psychology: General, 117*, 115-129.

Chignell, M. H. (1990, April). A Taxonomy of User Interface Terminology. *SIGCHI Bulletin*, p. 27-34.

Chignell, M. H., & Patty, B. W. (1987). Unidimensional scaling with efficient ranking methods. *Psychological Bulletin, 101*, 304-311.

Chignell, M. H., & Peterson, J. G. (1988a). Strategic issues in knowledge engineering. *Human Factors, 30*(4), 381-394.

Cody, W. J., Rouse, W. B., & Boff, K. R. (1993). *Automated information management for designers: Functional requirements for computer based associates that support access and use of technical information in design* (Technical Report AL/CF-TR-1993-0069). Wright Patterson Air Force Base, OH: Armstrong Laboratory.

Cohen, G. (1989). *Memory in the Real World*. Hillsdale, NJ: Lawrence Erlbaum.

Cohen, H. D., Bromage, R. C., Chinnis, J. J. O., Payne, J. W., & Ulvila, J. W. (1982). *A personalized and prescriptive attack planning decision aid*. (Technical Report 82-4). Falls Church, VA: Decision Science Consortium, Inc.

Cohen, M. S. (1993). Three paradigms for viewing decision biases. In G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), *Decision Making in Action: Models and Methods*. Norwood, NJ: Ablex.

Cohen, M. S., Leddo, J. M., & Tolcott, M. A. (1989). *Personalized and prescriptive aids for commercial air flight replanning* (Technical Report 89-2). Reston, VA: Decision Science Consortium, Inc.

Converse, S. A., & Kahler, S. E. (1992). *Knowledge acquisition and the measurement of shared mental models* (Report DAAL03-86-D-0001). Orlando, FA: Naval Air Warfare Center Training Systems Division.

Cook, T. D., & Campbell, D. T. (1979). *Quasi-experimentation*. Chicago, IL: Rand-McNally.

Coovert, M. D., Cannon-Bowers, J. A., & Salas, E. (1990). Applying mathematical modeling technology to the study of team training and performance. *Proceedings of The 12th Annual Interservice/Industry Training Systems Conference*. Orlando, Florida.

Coury, B. G., Motte, S., & Seiford, L. M. (1991). Capturing and representing decision processes in the design of an information system. *Proceedings of 35th Annual Meeting of the Human Factors Society* (pp. 1223-1227). Santa Monica, CA: Human Factors Society.

Crandall, B. W. (1989). A comparative study of think-aloud and critical decision knowledge elicitation methods. *SIGART Newsletter, Special Issue on Knowledge Acquisition*, p. 144-146.

Crandall, B. W., & Klein, G. (1990). The role of knowledge engineering in technology transfer. *Proceedings of IEEE 1990 National Aerospace and Electronics Conference NAECON* (pp. 878-880). New York: IEEE.

Crumley, L. M. (1989). *Review of research and methodologies relevant to army command and control performance measurement* (ARI Research Note 88-62 (AD A197 449)). Alexandria, VA: US Army Research Institute.

Crumley, L. M., & Sherman, M. B. (1990). *Review of command and control models and theory* (Technical Report 915 (AD A230 105)). Alexandria, VA: US Army Research Institute.

DeGreene, K. B. (1970). Systems analysis techniques. In K. B. DeGreene (Ed.), *Systems Psychology*. New York: McGraw-Hill.

Deimel, L. (1988). *User Interface Development* (Report SEI-SM-17.1.0). Pittsburg, PA: Carnegie Mellon University, Software Engineering Institute.

Department of Defense (1979). *Human engineering requirements for military systems, equipment and facilities* (Military Specification MIL-H-46855B). Washington, DC: Department of Defense.

Department of Defense (1988). *Military Standard: Defense System Software Development* (Military Specification DOD-STD-2167A). Washington, DC: Department of Defense.

Department of Defense (1991). *Military Standard on Task Performance Analysis* (Military Specification MIL-STD-1478). Washington, DC: Department of Defense.

Diaper, D. (1989). *Task Analysis for Human Computer Interaction*. Chichester: Ellis Horwood.

Dougherty, T. W., Ebert, R. J., & Callender, J. C. (1986). Policy capturing in the employment interview. *Journal of Applied Psychology, 71*(1), 9-15.

Downton, A., & Leedham, G. (1991). Human aspects of human-computer interface. In A. Downton (Ed.), *Engineering the Human-Computer Interface*. London: McGraw-Hill.

Drury, C. G., Paramore, B., Van Cott, H. P., Grey, S. M., & Corlett, E. N. (1987). Task Analysis. In G. Salvendy (Ed.), *Handbook of Human Factors*. New York: Wiley & Sons.

Duncan, K. D., & Shepherd, A. (1975). A simulator and training technique for diagnosing plant failures from control panels. *Ergonomics, 18*, 627-641.

Dzida, W. (1983). Das IFIP-Modell fur Benutzerschnittstellen. *Office Management, 31*, 6-8.

Elkerton, J., & Williges, R. C. (1989). Dialogue design for intelligent interfaces. In P. A. Hancock, & M. H. Chdignell (Eds.), *Intelligent Interfaces*. Amsterdam: North Holland.

Endsley, M. R. (1988). Design and evaluation for situation awareness enhancement. *Proceedings of 2nd Annual Meeting of the Human Factors Society* (pp. 96-106). Santa Monica, CA: Human Factors Society.

Engel, G. R., & Townsend, M. (1991). *Division command and control performance measurement* (Contract Report 01SE.W7711-7-7013). North York, Ontario: Defence and Civil Institute of Environmental Medicine.

Evans, J. (1988). The knowledge elicitation problem: A psychological perspective. *Behavior and Information Technology, 7*(2), 111-130.

Eysenck, M. W. (Ed.). (1990). *The Blackwell Dictionary of Cognitive Psychology.* Oxford, UK: Blackwell.

Eysenck, M. W., & Keane, M. T. (1990). *Cognitive Psychology: A Student's Handbook.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Fallesen, J. J. (1991). *Memoranda to Future Battle Laboratory concerning functional requirements on AirLan Battle Management (ALBM) Advanced Technology Transition Demonstration (ATTD).* Fort Leavenworth, KS: US Army Research Institute Field Unit.

Fallesen, J. J. (1993). *Overview of army tactical planning performance research* (Technical Report 984 (AD A273 273)). Alexandria, VA: US Army Research Institute.

Fallesen, J. J., & Quinkert, K. (1990). *Workspace design handbook for standardized command posts* (Research Product 90-29). Alexandria, VA: US Army Research Institute.

Farooq, M. U., & Diminick, W. D. (1988). A survey of formal tools and models for developing user interfaces. *International Journal of Man Machine Studies, 29,* 479-496.

Firley, M., & Hellens, D. (1991). *Knowledge Elicitation: A Practical Handbook.* London: Prentice Hall.

Fischer, G. (1990). Mental models (MM) - A computer scientist's point of view. In A. Turner (Ed.), *Mental Models and user-centered design.* ARI research note.

Flanagan, J. P. (1993). *Documentation of activities conducted in preparation for a knowledge base evaluation of the trend analysis and projection tool* (CAE-Link Report). Falls Church, VA: CAE-Link.

Fleishman, E. A., & Quaintance, M. K. (1984). *Taxonomies of Human Performance: The Description of Human Tasks.* Orlando, FL: Academic Press.

Foley, J. D. (1991). User interface software tools. *Proceedings of GI- 21 Jahrestagung* (pp. 3-18). Berlin: Springer-Verlag.

Galitz, W. O. (1989). *Handbook of Screen Format Design.* Wellesley, MA: QED, Information Sciences, Inc.

Gammock, J. G., & Young, R. M. (1985). Psychological techniques for eliciting expert knowledge. In A. Bramer (Ed.), *Research and Development in Expert Systems* (pp. 105-112). London: Cambridge University Press.

Gammock, J. (1987). Different techniques and different aspects on declarative knowledge. In A. Kidd (Ed.), *Knowledge Acquisition for Expert Systems* (pp. 137-163). New York: Plenum Press.

Gane, C., & Sarson, T. (1977). *Structured systems analysis: Tools and techniques* (Unpublished Report). McDonnell Douglas Corporation.

Gardner, A. (1991). An approach to formalized procedures for user-centered system design. In B. Shackel, & S. J. Richardson (Eds.), *Human Factors for Informatics Usability.* Cambridge: Cambridge University Press.

Gardner, K. (1990). Eliciting compiled knowledge in clinical pharmacology using "NEXTRA".

Proceedings of Second Annual Conference of the International Association of Knowledge Engineers (pp. 285-292).

Garg-Janardan, C., Eberts, R. E., Zimolong, B., Nof, S. Y., & Salvendy, G. (1987). Expert Systems. In G. Salvendy (Ed.), *Handbook of Human Factors*. New York: Wiley & Sons.

Garlinger, D. K., & Fallesen, J. J. (1988). *Review of command group training measurement methods* (Technical Report 798 (AD A201 753)). Alexandria, VA: US Army Research Institute.

Geiwitz, J., Kornell, J., & McCloskey, B. (1990). *An expert system for the selection of knowledge acquisition techniques* (Technical Report 785-2). Santa Barbara, CA: Anacapa Sciences.

Gentner, D., & Landers, R. (1985). Analogical reminding: A good representation of physics problems by experts and novices. *Cognitive Science, 5*, 121-152.

Gigerenzer, G. (1991). How to make cognitive illusions disappear: Beyond 'heuristics and biases'. *European Review of Social Psychology, 2*, 83-115.

Gordon, S. E. (1992). Implications of cognitive theory for knowledge acquisition. In R. R. Hoffman (Ed.), *The Psychology of Expertise: Cognitive Research and AI*. New York: Springer-Verlag.

Gordon, S. E., Schmierer, K. A., & Gill, R. T. (1993). Conceptual graph analysis: Knowledge acquisition for instructional system design. *Human Factors, 35*(3), 459-481.

Gould, J. D., & Lewis, C. (1985). Designing for usability: key principles and what designers think. *Communications of the ACM, 28*(3), 300-311.

Grant, S., & Mayes, T. (1991). Cognitive task analysis. In G. R. S. Weir, & J. L. Alty (Eds.), *Human Computer Interaction and Complex Systems*. London: Academic Press.

Grasser, A. C., & Gordon, S. E. (1991). Question answering and the organization of world knowledge. In W. Kesesen, A. Ortony, & F. Craik (Eds.), *Memories, Thoughts, and Emotions: Essays in Honor of George Mandler*. Hillsdate, NJ: Lawrence Erlbaum.

Green, M. (1985). Report on dialogue specification tools. In G. E. Pfaff (Ed.), *User Interface Management Systems* (pp. 9-20). Berlin: Springer-Verlag.

Greenberg, S., & Witten, I. H. (1985). Adaptive personalized interfaces - A question of viability. *Behavior and Information Technology, 4*(1), 31-45.

Greenfield, P. M. (1987). Electronic technologies, education, and cognitive development. In D. E. Berger, K. Pezdek, & W. P. Banks (Eds.), *Applications of Cognitive Psychology: Problem Solving, Education, and Computing*. Hillsdale, NJ: Lawrence Erlbaum.

Greeno, J. G., & Simon, H. A. (1986). Problem solving and reasoning. In R. C. Atkinson, R. J. Hermstein, G. Lindzey, & R. D. Luce (Eds.), *Handbook of Experimental Psychology*. New York: Wiley & Sons.

Grimes, J., & Potel, M. (1991). What is multimedia? *IEEE Computer Graphics & Application, January 1991*, 49-52.

Gruenberg, M. M., & Skyes, R. N. (1978). Knowledge and retention: The feeling of knowing and reminiscence. In M. M. Gruenberg, P. E. Morris, & R. N. Skyes (Eds.), *Practical Aspects of Memory*. London: Academic Press.

Halpin, S. M. (1992). *Army command and control evaluation system (ACCES): A brief description* (Working Paper LVN 92-01). Fort Leavenworth, KS: US Army Research Institute.

Hammond, K. R., Hamm, R. M., Grassia, J., & Pearson, T. (1987). Direct comparison of the efficacy of intuitive and analytical cognition in expert judgement. *IEEE transactions on Systems, Man, and*

Cybernetics, SMC-17(5), 753-770.

Hardy, E. J., & Klein, D. V. (1991). The Serpent UIMS. In H. J. Bullinger (Ed.), *Human Aspects in Computing: Design and Use of Interactive Systems and Work with Terminals*. Amsterdam: Elsevier.

Hartson, H. R., & Boehm-Davis, D. (1993). User interface development processes and methodologies. *Behaviour & Information Technology, 12*(2), 98-114.

Hartson, H. R., & Hix, D. (1989a). Human-computer interface development: Concepts and systems for its management. *ACM Computing Survey, 21*(1), 5-92.

Hartson, H. R., & Hix, D. (1989b). Towards empirically derived methodologies and tools for human-computer interface development. *International Journal of Man-Machine Studies, 31*, 477-494.

Hartson, H. R., & Smith, E. C. (1991). Rapid prototyping in human-computer interface development. *Interacting With Computers, 3*, 51-91.

Hashemi, S. (1990). PLEXSYS: An expert system development environment application in design support. *Proceedings of Proceedings of the Topical Meeting on Advances in Human Factors Research on Man/Computer Interactions: Nuclear and Beyond* (pp. 198-203). La Grange Park, IL: American Nuclear Society.

Henderson-Sellers, B., & Edwards, J. M. (1990). The object-oriented systems life cycle. *Communications of the ACM, 33*(9), 143-159.

Hix, D., & Schulman, R. S. (1991). Human-computer interface development tools: A methodology for their evaluation. *Communications of the ACM*, 34(3), 74-87.

Hobson, C. J., & Gibson, F. W. (1983). Policy capturing as an approach to understanding and improving performance appraisal: A review of the literature. *Academy of Management Review, 8*(4), 640-649.

Hoffman, R. R. (1989), A survey of methods for eliciting the knowledge of experts. *SIGART Newsletter number 108*, p. 19-27.

Hogarth, R. M. (1987). *Judgement And Choice* (2nd ed.). New York: Wiley & Sons.

Hollnagel, E. (1991). The influence of artificial intelligence on human-computer interaction: Much ado about nothing? In J. Rasmussen, H. B. Anderson, & N. O. Bernsen (Eds.), *Human-Computer Interaction, Research Directions in Cognitive Science, European Perspectives*. Hillsdale, NJ: Lawrence Erlbaum.

Hollnagel, E., Mancini, G., & Woods, D. D. (1986). *Intelligent Decision Support in Process Environments*. Berlin: Springer-Verlag.

Hollnagel, E., & Weir, G. (1988). Principles for dialogue design in man-machine systems. In J. Ranta (Ed.), *Analysis, Design, and Evaluation of Man-Machine Systems* (pp. 145-149). Oxford: Pergamon Press.

Holyoak, K. J., & Koh, K. (1987). Surface and structural similarity in analogical transfer. *Memory & Cognition, 15*, 332-240.

Holyoak, K. J., & Nisbett, R. E. (1986). Induction. In R. J. Sternberg, & E. Smith (Eds.), *The Psychology of Human Thought*. Cambridge: Cambridge University Press.

Holyoak, K. J., & Spellman, B. A. (1993). Thinking. In L. W. Porter & M. R. Rosenzweig (Eds.). *Annual Review of Psychology, 44*, 265-315.

Hopple, G. W. (1986). Decision-aiding dangers: The law of the hammer and other maxims. *IEEE Transactions of Systems, Man, and Cybernetics, SMC-16*(6), 948-964.

Hopson, J., & Zachary, W. (1982). Experiences in aiding airborne decision-making. *Society of Automotive Engineers Transactions, 91*, 4388-4402.

Hura, G. S. (1987). Petri net applications. *IEEE Potentials, October*, 25-28.

Hwang, C., & Yoon, K. (1981). *Multiple Attribute Decision Making: Methods and Applications. A State-of-the-Art Survey*. Berlin: Springer-Verlag.

Jacob, R. J. K., Legget, J. J., Myers, B. A., & Pausch, S. (1993). Interaction styles and input/output devices. *Behaviour & Information Technology, 12*(2), 69-79.

Jacob, V. S., Gaultney, L. D., & Salvendy, G. (1986). Strategies and biases in human decision making and their implications for expert systems. *Behaviour and Information Technology, 5*(2), 119-140.

Johnson, L., & Johnson, N. E. (1987). Knowledge elicitation involving teachback interviewing. In A. L. Kidd (Ed.), *Knowledge Acquisition for Expert Systems: A Practical Handbook* (pp. 91-108). New York: Plenum Press.

Johnson, P., & Johnson, H. (1989). Knowledge analysis of tasks: Analysis and specification for human-computer systems. In A. Downton (Ed.), *Engineering the Human-Computer Interface*. London: McGraw-Hill.

Johnson-Laird, P. N. (1989). Mental models. In M. I. Posner (Ed.), *Foundations of Cognitive Science* (pp. 470-499). London: Bradford.

Jones, E. R., & Grober, D. T. (1962). *The failure task analysis* (Report 8726). St. Louis, MO: McDonnell Aircraft Corp.

Kagel, A. S. (1986). The unshuffle algorithm. *Computer language, 3*(11), 61-66.

Kallela, J. (1992). Participative methods for design and development of process automation. *Proceedings of 11th European Annual Conference on Human Decision Making and Manual Control* (pp. 1-10). Valenciennes, France.

Kass, R., & Finn, T. (1991). General user modelling: A facility to support intelligent interaction. In J. W. Sullivan, & W. T. Sherman (Eds.), *Intelligent User Interfaces*. New York: ACM Press.

Kieras, D. E., & Polson, P. G. (1982). *An approach to the formal analysis of user complexity*. (Working Paper No. 2). University of Arizona, and University of Colorado: Project on the User Complexity of Devices and Systems.

Kirlik, A., Markert, W. J., & Kossack, M. (1992). *Comparison of display enhancement with intelligent decision-aiding* (Report NASA-CR-189846 (N92-17132)). Atlanta, GA: Georgia Institute of Technology.

Kirwan, B., & Ainsworth, L. K. (Eds.). (1992). *A Guide To Task Analysis*. London: Taylor and Francis.

Klein, G. A. (1987). Analytical versus recognitional approaches to design decision making. In W. B. Rouse, & K. R. Boff (Eds.), *System Design. Behavioural Perspectives On Designers, Tools, And Organizations* (pp. 175-186). Amsterdam: North Holland.

Klein, G. A. (1993). *Naturalistic decision making: Implications for design* (CSERIAC SOAR 93-1). Dayton, Ohio: Crew System Ergonomics Research Information Analysis Center.

Klein, G. A., Calderwood, R., & MacGregor, D. (1989). Critical decision method for eliciting knowledge. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-19*(3), 462-472.

Klein, G. A., Orasanu, J., Calderwood, R., & Zsambok, C. E. (Eds.). (1993). *Decision Making in Action: Models and Methods*. Norwood, NJ: Ablex Publishing Corp.

Koller, F. (1992). Multimedia interfaces. In M. Galer, S. Harker, & J. Ziegler. (Eds.), *Methods and Tools in*

User-Centered Design for Information Technology. Amsterdam: Elsevier.

Kosslyn, S. M. (1980). *Image and Mind*. Cambridge, MA: Harvard University Press.

Kuhme, T., Hornung, G., & Witschital, P. (1991). Conceptual models in the design process of direct manipulation user interfaces. In H. J. Bullinger (Ed.), *Human Aspects in Computing: Design and Use of Interactive Systems and Work with Terminals* (pp. 722-727). Amsterdam: Elsevier.

Lachman, J. L., Lachman, R., & Thornsberry, C. (1979). Metamemory through the adult lifespan. *Developmental Psychology, 15*(5), 543-551.

Lane, N. E. (1986). *Issues in performance measurement for military aviation with applications to air combat maneuvering* (Report NTSC TR-86-008). Orlando, FL: Naval Training Systems Center.

Laughery, S. K. R., & Laughery, S. S. R. (1987). Analytic techniques for function analysis. In G. Salvendy (Ed.), *Handbook of Human Factors*. New York: Wiley & Sons.

Leddo, J., Cohen, M. S., O'Connor, M. F., Bresnick, T. A., & Marvin, F. F. (1991). *Integrating knowledge elicitation and representation framework* (Technical Report ARI-TR-927 (AD-A239 843/6/XAB)). Alexandria, VA: US Army Research Institute.

Lee, W. (1971). *Decision Theory and Human Behaviour*. New York: Wiley & Sons.

Leplat, J., & Bisseret, A. (1965). Analyse des processus de traitement de l'information chez le controleur de la navigation aerienne. *Bulletin du C.E.R.P., 1-2*, 51.

Lewis, C. H. (1990). A research agenda for the nineties in human-computer interaction. *Human-Computer Interaction, 5*, 125-143.

Lipshitz, R. (1993). Decision making as argument driven action. In G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), *Decision Making in Action: Models and Methods*. Norwood, N.J.: Ablex.

Lussier, J. W. (1991). *Early decisions and concurrent option comparison in problem solving groups* (Research Report 1618). Alexandria, VA: US Army Research Institute.

Lysaght, R. J., Hill, S. G., Dick, A. O., Plamondon, B. D., Linton, P. M., Wierville, W. W., Zaklad, A. L., Bittner, A. C., & Wherry, R. J. (1989). *Operator workload: Comprehensive review and evaluation of operator workload methodologies* (Technical Report 851 (Also AD A212 879)). Alexandria, VA: Army Research Institute.

Mackie, R. R., & Wylie, C. D. (1985). *User Considerations in the Acceptance and Use of AI Decision Aids*. Goleta, CA: Essex Inc.

MacMillan, J., Entin, E. E., & Lentz, P. (1988). *Experiment report: The effects of option planning and battle workload on command and control effectiveness* (Technical Report TR-368). Burlington, MA: Alphatech Inc.

Malhotra, N. (1982). Information load and consumer decision making. *Journal of Consumer Research, 8*, 419-429.

Malinowski, U., Kuhme, T., Dietrich, H., & Schneider-Hufschmidt (1992). A taxonomy of adaptive user interfaces. *Proceedings of HCI*. Cambridge: University Press.

Marcus, A., & Van Dam, A. (1991). User-interface developments for the nineties. *IEEE Computer, 24*(9), 49-57.

Marshall, C., Nelson, C., & Gardiner, M. M. (1987). Design guidelines. In M. M. Gardiner, & B. Christie (Eds.), *Applying Cognitive Psychology to User-Interface Design*. New York: John Wiley & Sons.

Mayer, R. E. (1981). The psychology of how novices learn computer programming. *ACM Computing*

Surveys, 13, 1-12.

McCann, C. A. (1989). *RSG.12 on Computer-Human Interaction in Command and Control* (Final Report AC/243(P8/RSG.12)D/7). Brussels, Belgium: NATO HQ, Defence Research Section.

McCann, C. A. (1990). *Human cognitive processes in command and control planning. 1: Identification of research issues* (Report 1990 B-18). Soesterberg, NL: TNO Human Factors Research Institute.

McCann, C. A., & Essens, P. J. M. D. (1991). *Human cognitive processes in command and control planning. 3: Determining basic processes in planning in time and space* (Report 1991 B-11). Soesterberg, NL: TNO Human Factors Research Institute.

McCloskey, B. P., Geiwitz, J., & Kornell, J. (1991). Empirical comparisons of knowledge acquisition techniques. *Proceedings of Human Factors Society 35th Annual Meeting* (pp. 268-272). Santa Monica, CA: Human Factors Society.

McDonald, I., Dearholt, D., Paap, K., & Schvaneveldt, R. (1986). A formal interface design methodology based on user knowledge. *Proceedings of CHI '86* (pp. 285-290). New York : Association for Computing Machinery.

McNeese, M. D., & Zaff, B. S. (1991). Knowledge as design: A methodology for overcoming knowledge acquisition bottlenecks in intelligent interface design. *Proceedings of Human Factors Society 35th Annual Meeting* (pp. 1181-1185). Santa Monica, CA: Human Factors Society.

Meachan, J. A., & Emont, N. C. (1989). The interpersonal basis of everyday problem solving. In J. D. Sinnnott (Ed.), *Everyday Problem Solving*. New York: Praeger.

Meister, D. (1985). *Behavioral Analysis and Measurement Methods*. New York: Wiley & Sons.

Meister, D. (1991). *Psychology of System Design*. Amsterdam: Elsevier.

Merkelbach, E. J. H. M., & Schraagen, J. M. C. (1994). *A framework for the analysis of cognitive tasks* (Report 94-B13). Soesterberg, NL: TNO Human Factors Research Institute.

Merriman, S. C., Muckler, F., Howells, H., Olive, B. R., & Beevis, D. (1985). *Workshop on applications of systems ergonomics to weapons system development* (Report DS/A/DR(84)408). Brussels, Belgium: NATO HQ, Defence Research Section.

Metersky, M. L. (1993). A decision-oriented approach to system design and development. *IEEE Transactions of Systems, Man, and Cybernetics, 23*(4), 1024-1037.

Meyer, B., Yakemovic, K. C. B., & Harris, M. (1993). Issues in practical application of an adaptive interface. *Proceedings of Intelligent User Interfaces '93* (pp. 251-254). Baltimore, MD: Association for Computing Machinery.

Miles, J. J. L., & Geddie, J. C. (1989). *Proposed military standard. MIL-STD-TASK. Task Analysis* (Working Paper MSG 89-01). Fort Leavenworth, KA: US Army Research Institute, Fort Leavenworth Field Unit.

Miller, R. B. (1971a). *Development of a taxonomy of human performance: Design of a systems task vocabulary.* (Research Study 71-4). Arlington, VA: Behavior and Systems Research Laboratory.

Mitchell, A. A. (1987). The use of alternative knowledge-acquisition procedures in the development of a knowledge-based media planning system. *International Journal of Man-Machine Studies, 26,* 399-411.

Mitta, D. A. (1989). Knowledge acquisition: Human Factors issues. *Proceedings of 33rd Annual Meeting of the Human Factors Society* (pp. 351-355). Santa Monica, CA: Human Factors Society.

Monarchi, D. E., & Puhr, G. I. (1991). A research typology for object-oriented analysis and design. *Communications of the ACM, 35*(9), 35-47.

Montgomery, H. (1993). The search for a dominance structure in decision making: Examining the evidence. In G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), *Decision Making in Action: Models and Methods* (pp. 182-187). Norwood, NJ: Ablex.

Moran, T. P. (1981). The Command Language Grammar: A representation for the user interface of interactive computer systems. *International Journal of Man-Machine Studies, 15*, 3-50.

Morris, N. M. (1987). Designing for user acceptance of design aids. In W. B. Rouse, & K. R. Boff (Eds.), *System Design: Behavioral Perspectives on Designers, Tools, and Organizations*. Amsterdam: North-Holland.

Mrdalj, S. (1990). Bibliography of object-oriented system development. *ACM SIGSOFT Software Engineering Notes, 15*(5), 60-63.

Murphy, E. D., & Mitchell, C. M. (1986). Cognitive attributes: implications for display design in supervisory control systems. *International Journal Man-Machine Studies, 25*, 411-438.

Murray, D. M. (1987). Embedded user models. In H. J. Bullinger, & B. Shackel (Eds.), *Proceedings of Human-Computer Interaction-INTERACT '87* (pp. 229-235). Amsterdam: Elsevier.

Myers, B. A. (1989). User-Interface Management Control and Communication. *IEEE Software, 6*(1), 15-23.

Myers, B. A., & Rosson, M. B. (1992). Survey on user interface programming. Human factors in computing systems - Striking a balance. *Proceedings of CHI '92* (pp. 195-202). New York: Association for Computing Machinery Press.

Neilson, I. (1987). *Techniques for specifying dialogue.* (AMU8729/01S). Glasgow: Scottish HCI Center.

Newell, A., Rosenbloom, P. S., & Laird, J. E. (1989). Symbolic architectures for cognition. In M. I. Posner (Ed.), *Foundations of Cognitive Science* (pp. 92-131). London: Bradford.

Newman, W., & Sproull, R. (1979). *Principles of Interactive Computer Graphics* (2nd ed.). New York: McGraw-Hill.

Nickerson, R. S., & Feeher, C. E. (1975). *Decision making and training: A review of theoretical and empirical studies of decision making and their implications for the training of decision makers* (Technical Report NAVTRAEQUIPCEN 73-C-0128-1). Cambridge, MA: Bolt, Beranek, and Newman.

Nickerson, R. S., & Pew, R. W. (1990). Toward more compatible human-computer interfaces. *IEEE Spectrum, 27*, 40-43.

Noble, D. (1993). A model to support development of situation assessment aids. In G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), *Decision Making in Action: Models and Methods* (pp. 287-305). Norwood, NJ: Ablex.

Noble, D., Grosz, C., & Boehm-Davis, D. (1987). *Rules, schema and decision making* (Report R-125-87). Vienna, VA: Engineering Research Associates.

Noble, D. F. (1989). Schema-based knowledge elicitation for planning and situation assessment aids. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-19*(3), 473-482.

Norcio, A. F., & Stanley, J. (1989). Adaptive human-computer interfaces: A literature survey and perspective. *IEEE Transactions on Systems, Man, and Cybernetics, 19*, 399-408.

Norman, D. A. (1981). The trouble with UNIX. *Datamation, 27*, 139-150.

Norman, D. A. (1983). Some observations on mental models. In D. Genter, & A. L. Stevens (Eds.), *Mental Models*. Hillsdale, N.J.: Lawrence Erlbaum.

Norman, D. A. (1985). Twelve issues for cognitive science. In A. M. Aitkenhead, & J. M. Slack (Eds.), *Issues in Cognitive Modelling*. London: Lawrence Erlbaum.

Norman, D. A. (1986). New views of information processing: Implications for intelligent systems support. In E. Hollnagel (Ed.), *Intelligent Decision Support in Process Environments*. Heidelberg: Springer-Verlag.

Norman, D. A. (1988). *The Psychology of Everyday Things*. New York: Basic Books.

Norman, D. A., & Draper, S. W. (1986). *User Centered System Design: New Perspectives on Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum.

Orasanu, J., & Connolly, T. (1993). The reinvention of decision making. In G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), *Decision Making in Action: Models and Methods* (pp. 3-20). Norwood, NJ: Ablex.

Overmyer, S. P. (1990). The impact of DOD-STD-2167A on iterative design methodologies: Help or hinder. *ACM SIGSOFT, Software Engineering Notes, 15*(5), 50-59.

Papazian, B., Roberts, R. B., Redick, D. J. A., Tani, D. M., Pew, R. W., & Salter, W. S. (1989). *An intelligent tool for the design of presentations: A system identification study* (Report RADC-TR-89-197). Griffiss AFB, NY: Rome Air Development Center.

Payne, J. R., Braunstein, T. J., Ketchel, J. M., & Pease, M. C. (1975). *A brief survey of potential decision aids for the task force commander and his staff* (Report NWRC-RM-84). Palo Alto, CA: Stanford Research Institute.

Payne, J. W., Bettman, J. R., & Johnson, E. J. (1988). Adaptive strategy selection in decision making. *Journal of Experimental Psychology: Leaning, Memory and Cognition, 14*, 534-552.

Payne, S. J., & Green, T. R. G. (1989). Task-action grammar: The model and its developments. In D. Diaper (Ed.), *Task Analysis for Human Computer Interaction* (pp. 75-107). Chichester: Ellis Horwood.

Pennington, N., & Hastie, R. (1993). A theory of explanation-based decision making. In G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), *Decision Making in Action: Models and Methods* (pp. 188-204). Norwood, NJ: Ablex.

Perkins, D., Jay, E., & Tishman, S. (1993). New conceptions of thinking: From ontology to education. *Educational Psychologist, 28*, 67-85.

Perlman, G. (1988). Software tools for user interface development. In M. Helander (Ed.), *Handbook of Human-Computer Interaction*. Amsterdam: Elsevier.

Perlman, G. (1989). *User Interface Development* (Curriculum Module SEI-CM-17-1). Pittsburg, PA: Carnegie-Mellon University, Software Engineering Institute.

Phelps, R. H., Halpin, S. M., & Johnson, E. M. (1981). *A decision support framework for decision aid designers* (Technical Report 504). Alexandria, VA: US Army Research Institute.

Plott, C., Dahl, S., Laughery, R., & Dick, A. (1989). *Concept analysis for simulation modification methodology* (Research Report 613 (AD A250 947)). Alexandria, VA: US Army Research Institute.

Posner, M. I. (1989). *Foundations of Cognitive Science*. Cambridge, MA: MIT Press.

Pounds, J., & Fallesen, J. J. (1994). *Understanding problem solving strategies* (Technical Report in preparation). Alexandria VA: Army Research Institute.

Powell, G., & Schmidt, C. (1988). *A first-order computational model of human operational planning*. Fort Monmouth, NJ: Center for Command, Control and Communications.

Puckett, T. R. (1990). *Rule-based expert systems in the command estimate: An operational perspective.* Master's thesis, US Army Command and General Staff College, Ft. Leavenworth, KS.

Quillian, M. R. (1966). *Semantic memory*. Doctoral dissertation, Carnegie Institute for Technology, Pittsburg, PA.

Ramsey, H. R., & Atwood, M. E. (1979). *Human factors in computer systems: A review of the literature.* (Technical Report SAI-79-111-DEN). Englewood, CO: Science Applications, Inc.

Rasmussen, J. (1982). Human errors. A taxonomy for describing human malfunction in industrial installations. *A Journal of Occupational Accidents*, 4, 311-333.

Rasmussen, J. (1986). *Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering*. Amsterdam: Elsevier Science.

Rasmussen, J. (1993). Deciding and doing: Decision making in natural contexts. In G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), *Decision Making in Action: Models and Methods* (pp. 158-171). Norwood, NJ: Ablex.

Read, J. D., & Bruce, D. (1982). Longitudinal tracking of difficult memory retrievals. *Cognitive Psychology*, 14, 230-300.

Reason, J. (1990). *Human Error*. Cambridge, UK: Cambridge University Press.

Redding, R. E. (1989). Perspectives on cognitive task analysis: the state of the state of the art. *Proceedings of Human Factors Society 33rd Annual Meeting* (pp. 1348-1352). Santa Monica, CA: Human Factors Society.

Riedel, S. L. *Design and management of life cycle user evaluations of decision support systems* (Unpublished Report). Fort Leavenworth, KA: US Army Research Institute, Fort Leavenworth Research Unit.

Riedel, S. L. (1988). *User acceptance and field implementation of decision support systems* (Research Report 1477). Fort Leavenworth, KA: US Army Research Institute, Fort Leavenworth Field Unit.

Riekert, W. (1991). Knowledge acquisition as an object-oriented modelling process. In M. J. Tauber, & D. Ackermann (Eds.), *Mental Models and Human Computer Interaction* (pp. 373-381). Amsterdam: Elsevier Sciences.

Rips, L. J. (1986). Deduction. In R. J. Sternberg, & E. Smith (Eds.), *The Psychology of Human Thought.* Cambridge: Cambridge University Press.

Rodi, L. L., Pierce, J. A., & Dalton, R. E. (1989). Putting the expert in charge: Graphical knowledge acquisition for fault diagnosis and repair. *SIGART Newsletter number 108*, p. 56-62.

Rook, F. W. (1984). *Human factors guidelines for display interface design: Senior battle staff decision aids* (Report F30602-84-C-0154). New Hartford, NY: Par Technology.

Rosch, E. (1978). Principles of categorization. In E. Rosch, & B. B. Loyd (Eds.), *Cognition and Categorization*. Hillsdale, NJ: Lawrence Erlbaum.

Roth, E. M., Woods, D. D., & Pople, H. E., Jr. (1992). Cognitive simulation as a tool for cognitive task analysis. *Ergonomics*, 35, 1163-1198.

Rouse, W. B. (1981). *Systems Engineering Models of Human-Machine Interaction*. New York : North Holland.

Rouse, W. B. (1988). Adaptive aiding for human/computer control. *Human Factors*, 30(4), 431-443.

Rouse, W. B. (1990). Designing for human error: Concepts for error tolerant systems. In H. R. Booher (Ed.), *MANPRINT: An Approach to Systems Integration*. New York: Van Nostrand Reinhold.

Rouse, W. B. (1991). *Design for Success - A Human Centered Approach to Designing Successful Products and Systems.* New York: Wiley & Sons.

Rouse, W. B., & Boff, K. R. (1987a). Designers, tools, and environments: State of knowledge, unresolved issues, and potential directions. In W. B. Rouse, & K. R. Boff (Eds.), *System Design: Behavioral Perspectives on Designers, Tools and Organizations* (pp. 43-63). Amsterdam: North-Holland.

Rouse, W. B., & Boff, K. R. (1987b). Workshop themes and issues: The psychology of system design. In W. B. Rouse, & K. R. Boff (Eds.), *System Design: Behavioral Perspectives on Designers, Tools and Organizations* (pp. 7-17). Amsterdam: North-Holland.

Rouse, W. B., Geddes, N. D., & Curry, R. E. (1987). An architecture for intelligent interfaces: Outline of an approach to supporting operators of complex systems. *Human-Computer Interaction, 3*, 87-122.

Rouse, W. B., & Morris, N. M. (1986). On looking into the black box: Prospects and limits in the search for mental models. *Psychological Bulletin, 100*(3), 349-363.

Rouse, W. B., & Rouse, S. H. (1983). Analysis and classification of human error. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-13*, 539-549.

Rumelhart, D. E. (1977). *Introduction to Human Information Processing.* New York : John Wiley & Sons.

Rumelhart, D. E. (1989). The architecture of mind: A connectionist approach. In M. I. Posner (Ed.), *Foundations of Cognitive Science.* Cambridge, MA: MIT Press.

Ryder, J. M., & Redding, R. E. (1993). Integrating cognitive task-analysis into instructional-systems development. *Educational Technology Research and Development, 41*(2), 75-96.

Sage, A. P. (1991). *Decision Support Engineering.* New York: Wiley & Sons.

Sage, A. P. (1992). *Systems Engineering.* New York: Wiley & Sons.

Sage, A. P., & Rouse, W. B. (1986). Aiding the human decision maker through the knowledge-based sciences. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-16*, 511-521.

Salvendy, G. (1987). *Handbook of Human Factors.* New York: Wiley & Sons.

Samet, M. G. (1984). Computer-based aids for tactical information processing and decision making: Selected developments and guidelines for design, evaluation, and selection. *Proceedings of ARI Command and Control Research Planning Workshop.* San Francisco, CA: unpublished report.

Schaafstal, A. M., & Schraagen , J. M. C. (1992). *A method for cognitive task analysis* (Report 1992 B-5). Soesterberg, NL: TNO Human Factors Research Institute.

Schank, R. C. (1985a). *Dynamic Memory.* Cambridge: Cambridge University Press.

Schank, R. C. (1985b). Reminding and memory organization. In A. M. Aitkenhead, & J. M. Slack (Eds.), *Issues in Cognitive Modelling* (pp. 15-62). Hillsdale, NJ: Lawrence Erlbaum.

Schvanevelt, R. W., Durso, F. T., Goldsmith, T. E., Breen, T. J., Cooke, N. M., Taucker, R. G., & Maio, J. C. d. (1985). Measuring the structure of expertise. *International Journal of Man-Machine Studies, 23*, 699-728.

Schweickert, R., & Fisher, D. L. (1987). Stochastic Network Models. In G. Salvendy (Ed.), *Handbook of Human Factors.* New York: John Wiley.

Shackel, B. (1984). The concept of usability. In J. Bennet, D. Case, J. Sandelin, & M. Smith (Eds.), *Visual Display Terminals.* Englewood Cliffs, NJ: Prentice-Hall.

Shadbolt, N., & Burton, A. M. (1989), The Empirical Study of Knowledge Elicitation Techniques. *SIGART Newsletter number 108*, p. 15-18.

Simon, H. A. (1956). Rational choice and the structure of the environment. *Psychological Review, 63,* 129-138.

Simon, H. A., & Kaplan, C. A. (1989). Foundations of cognitive science. In M. I. Posner (Ed.), *Foundations of Cognitive Science*. London: Bradford.

Smith, E. E. (1988). Concepts and thought. In R. J. Sternberg, & E. E. Smith (Eds.), *The Psychology of Human Thought* (pp. 502-526). Cambridge: Cambridge University Press.

Smith, S. L., & Mosier, J. N. (1986). *Guidelines for designing user interface software* (Technical Report ESD-TR-278). Hanscom AFB, MA: Electronic Systems Division.

Sprague, & Carlson (1982). *Building Effective Decision Support Systems*. Engelwood Cliffs, NJ: Prentice-Hall.

St. Denis, G. (1990). *How to facilitate the process of translating system requirements into a virtual prototype: VAPS, the design process, and human engineering* (Final Report 03SE.W7711-8-7049). Montreal, Quebec: Virtual Prototypes Inc.

Sullivan, J. W., & Sherman, W. T. (1991). *Intelligent User Interfaces*. New York: ACM Press.

Swaffield, G., & Knight, B. (1990). Applying the system analysis techniques to knowledge engineering. *Expert Systems, 7,* 82-93.

Thompson, J. R., Hopf-Weichel, R., & Geiselman, R. E. (1984). *The cognitive bases of intelligence analysis* (Research Report 1362). Alexandria, VA: US Army Research Institute.

Thordsen, M., Galushka, J., Klein, G. A., Young, S., & Brezovic, C. P. (1989). *Knowledge elicitation study of military planning* (Technical Report 876 (AD A219 969)). Alexandria, VA: US Army Research Institute.

Thordsen, M., & Klein, G. (1991). Cognitive features of team decision making. *Proceedings of IEEE 1990 National Aerospace and Electronics Conference NAECON* (pp. 891-893). New York : IEEE.

Thovtrop, H., & Nielsen, J. (1991). *Assessing the usability of a user interface standard* (Report 0-89791-383-3/91/0004/). New York : Association for Computing Machinery.

Tolcott, M. A., & Holt, V. E. (1987). *Impact and potential of decision research on decision aiding* (Technical Report (AD-A192 276/4/XAB)). Arlington, VA: Office of Naval Research.

Tullis, T. S. (1985). Designing a menu-based interface to an operating system. *Proceedings of Human Factors in Computing Systems (CHI)* (pp. 79-84). New York : Association for Computing Machinery.

Tulving, E. (1972). Episodic and semantic memory. In E. Tulving (Ed.), *Organization of Memory*. New York: Academic Press.

Tulving, E., & Thomson, D. M. (1973). Encoding specificity and retrieval processes in episodic memory. *Psychological Review, 80,* 352-373.

Tversky, A. (1972). Elimination by aspects: A theory of choice. *Psychological Review, 79*(4), 281-299.

VanLehn, K. (1989). Problem solving and cognitive skill acquisition. In M. I. Posner (Ed.), *Foundations of Cognitive Science* (pp. 527-579). London: Bradford.

Visser, W., & Morais, A. (1991). Concurrent use of different expertise elicitation methods applied to the study of programming activity. In M. J. Tauber, & D. Ackermann (Eds.), *Mental Models and Human Computer Interaction* (pp. 97-113). Amsterdam: Elsevier .

Walsh, P. (1989). Analysis for task object **modelling (ATOM)**: Towards a method of integrating task **analysis** with Jackson System Development for user interface software design. In D. Diaper (Ed.), *Task Analysis for Human Computer Interaction* (pp. 186-209). Chichester: Ellis Horwood.

Wassermann, A. I. (1980). Information system design technology. *Journal of the American Society for Information Sciences, Jan*, 5-24.

Watermann, D. A. (1986). *A Guide to Expert Systems*. Reading, MA: Addison-Wesley.

Weber, E. U., & Coskunoglu, O. (1990). Descriptive and prescriptive models of decision making: Implications for the development of decision aids. *IEEE Transactions of Systems, Man, and Cybernetics, 20*(2), 310-317.

Weingaertner, S. T., & Lewis, A. H. (1988). Evaluation of decision aiding in submarine emergency decision making. In J. Ranta (Ed.), *Analysis, Design and Evaluation of Man-Machine Systems: Selected Papers from the 3rd IFAC/IFIP/IES/IFORS Conference* (pp. 195-201). Oxford, UK: Pergamon Press.

Whaley, C. P. (1979). Collecting paired-comparison data with a sorting algorithm. *Behavior Research Methods and Instrumentation, 11*, 147-150.

Whitefield, A. (1991). Human-computer interaction models and their roles in the design of interactive systems. In P. Falzon (Ed.), *Cognitive Ergonomics: Understanding, Learning and Designing Human-Computer Interaction*. London: Academic Press.

Wickens, C. D. (1984). *Engineering Psychology and Human Performance*. Columbus, OH: Charles E. Merrill Publishing Company.

Williams, J. R. (1989). Guidelines for dialogue design. In G. Salvendy, & M. J. Smith (Eds.), *Designing and Using Human-Computer Interfaces and Knowledge Based Systems* (pp. 589-596). Amsterdam: Elsevier.

Williges, R. C. (1987). The use of models in human-computer interface design. *Ergonomics, 30*(3), 491-502.

Williges, R. C., Williges, B. H., & Elkerton, J. (1987). Software interface design. In G. Salvendy (Ed.), *Handbook of Human Factors*. New York: Wiley & Sons.

Wilson, J., & Rosenberg, D. (1988). Rapid prototyping for user interface design. In M. Helander (Ed.), *Handbook of Human-Computer Interaction*. Amsterdam: North Holland.

Wilson, J. R., & Rutherford, A. (1989). Mental models: Theory and application in human factors. *Human Factors, 31*(6), 617-634.

Witus, G., Patton, J., & Cherry, P. (1985). *Automated assistance for fire support command, control, communications, and intelligence (C3I)* (Research Product 85-1). Alexandria, VA: US Army Research Institute.

Wohl, J. G., Entin, E. E., Kleinman, D. L., & Pattipati, K. (1984). Human decision processes in military command and control. In *Advances in Man-Machine Systems Research* (pp. 261-307). Greenwich, CT: JAI Press.

Woods, D. D. (1991). The Cognitive Engineering of Problem Representation. In G. R. S. Weir, & J. L. Alty (Eds.), *Human-Computer Interaction and Complex Systems* (pp. 169-188). London: Academic Press.

Woods, D. D., & Hollnagel, E. (1987). Mapping cognitive demands in complex problem-solving worlds. *International Journal of Man-Machine Studies, 26*, 257-275.

Woods, D. D., & Roth, E. M. (1988). Cognitive Systems Engineering. In M. Helander (Ed.), *Handbook of Human-Computer Interaction* (pp. 1-41). Amsterdam: Elsevier.

Woodward, B. (1990). Knowledge acquisition at the front end: Defining the domain. *Knowledge Acquisition*, 2, 73-94.

Zachary, W. (1986). A cognitively based functional taxonomy of decision support techniques. *Human-Computer Interaction*, 2, 25-63.

Zachary, W. (1988). Decision support systems: Designing to extend the cognitive limits. In M. Helander (Ed.), *Handbook of Human-Computer Interaction* (pp. 997-1030). Amsterdam: Elsevier.

Zachary, W. W., & Ross, L. (1991). Enhancing human-computer interaction through use of embedded COGNET models. In *Human Factors Society 35th Annual Meeting*. Santa Monica, CA: Human Factors Society.

Zaklad, A. L., Bulger, J. P., Glenn, F. A., Zachary, W. W., & Hicinbothom, J. (1986). *Decision aids for the FAAD ABMOC* (Technical Report 2018). Willow Grove, PA: Analytics.

Zsambok, C. E., Beach, L. R., & Klein, G. A. (1992). *A literature review of analytical and naturalistic decision making (DRAFT)* (Technical Report Task 2). San Diego, CA: Naval Command, Control and Ocean Surveillance Center.

This page has been left blank intentionally

APPENDIX A. CONCEPTS USED IN MODELLING COGNITION

A.1. Introduction
A.1.1. Overview of Approaches to Cognitive Modelling
A.1.2. Fundamental Characteristics of Cognitive Entities
A.2. Elements of the Cognitive System
A.2.1. Memory
A.2.2. Representation of Knowledge
A.2.2.1. Types and Organisation of Knowledge
A.2.2.2. Knowledge on Concepts and Relations
A.2.2.3. Knowledge on Concepts and Relations
A.2.2.4. Hybrid Representations and Mental Models
A.2.2.5. Summary
A.2.3. Basic Processes and Operations in Cognition
A.2.4. Meta-cognition
A.2.5. Learning
A.3. Schema-Based Model of Problem Solving
A.4. Implications for COADE

A.1 INTRODUCTION

1. The processes and structures of cognition[3] define how we think about problems and what difficulties we encounter in solving them. At its most general level, cognition includes all thinking. Why are we so interested in cognition for COADE? It is because cognitive structures and processes are the "basic mechanisms" of human performance. If the aim is to support and enhance problem solving and decision performance[4], then we need to better understand these mechanisms, how they may predispose a human to errors, and how they fit with enhancements.

2. The typical approach to analysing human behaviour in the past has been task analysis. Historically, task analysis has emphasised environmental and task variables. The approaches have not necessarily included the cognitive aspects of task performance. Yet decision making tasks typically place high demands on cognitive skills, since they are complex, dynamic, filled with uncertainty, and knowledge-rich (requiring large amounts of experience). The essence of performance in C2 decision making tasks lies in how the decision maker develops an understanding of a problem, creates and tests solutions, and learns from the implementation of the chosen course of action. Thus it is important to have a cognitive basis for conceptualising and studying C2 tasks, as a basis for a cognitive task analysis. This permits a cognitive model of the task to be constructed. A cognitive model makes explicit the cognitive structure and operations involved in a person's knowledge representation and mental processing in carrying out a task. It describes how a person's world is represented and what processes are used to manipulate that world conceptually.

3. The particular cognitive modelling approach that is used in a specific COADE application will largely determine what aspects of cognition are emphasised. Since cognition is not observable by direct means, our beliefs about it are critical for successfully shaping our search and understanding about performance. It is an implicit assumption in COADE that the potential for improving task or job

[3] Concepts relating to cognition and cognitive modelling are underlined in this section.
[4] We do not make a strong distinction between decision making, planning and problem solving in this section, preferring instead to consider these activities as different ways of viewing cognition.

performance and problem solving behaviour will depend on the specifics of the task, job, and required mental skills. Various analyses are suggested in COADE to assist the decision aid developer to better understand the task, situational, and organisational dependencies. Cognitive modelling helps to address practical aiding questions on cognitive demands and likely errors. The model helps to structure the different aspects of cognition so that different types of processing and different types of errors are considered.

4. This Appendix reviews several general ways of viewing cognition. COADE advocates can use these different views, concepts or theories to develop specific cognitive models for particular decision situations.

5. To support the importance of cognitive models, we can refer to Lewis (1990) who contends,

> "It is perhaps less clear that working on cognitive theory offers anything to people who do not see theory as crucial . . . . Consider, for example, a process-is-paramount person, sceptical about the value of theoretical abstractions. Such a person doubts that inquiries into associative processes, whatever their outcome, will contribute anything to the empirical iterative design process he or she relies on. But it can be argued that this person is actually relying on a covert cognitive theory and will benefit from improvements in it. This covert theory can be seen at work at two points in the iterative design cycle. First, the analysis of user errors, which is fundamental to iterative design depends on identifying and discriminating cognitive processes. For example, it is important to know whether an error results from incorrect actions in pursuit of an appropriate goal or from pursuit of an inappropriate goal. Second, the choice of remedy for problems invokes theory in the same way. What changes to this prompt will make it more likely to be comprehended accurately or to be read at all? An answer presupposes a theoretical analysis." (p 135)

6. In more specific terms, beyond description and prescription, cognitive models offer expectations about general and idiosyncratic behaviours. For example, the concept of "scripts" allow for the filling in or imagining of details when we think about an event or situation. Scripts guide and enrich our understanding and provide an organising framework for remembering events.

7. Cognitive science does not yet portray a complete or integrated picture of perception, memory, concept formation, attention, decision making, language, and action; much yet remains to be learned about how people think and solve problems. However, cognitive science takes a broader view of human performance than either behaviourist or data-driven information processing approaches by regarding humans as goal-directed beings having cooperative mental capacities.

8. Much of the research on modelling cognition has been stimulated by debate about the merits of one theory over other. Although this has lead to increased understanding, it could lead to the assumption that one theoretical basis is better than another. Most theories emphasise a specific aspect of cognition and do not aspire to be a unified theory of cognition. COADE users will have difficulty if they are concerned with the accuracy of the theories or concepts; the actual criterion should be the usefulness of the modelling approach.

9. This chapter does not intend to resolve any of the great debates about modelling cognition. Its aims are much more practical: to provide an overview of some of the concepts used in cognitive modelling; address similarities among theoretical viewpoints; consider relevant research findings; and to provide implications for COADE activities.

10. With the broader background in cognition and thinking provided by this Appendix, the user of COADE will be better able to appreciate human cognitive performance — both its limitations and styles — and will be in a better position to understand possibilities for performance aiding and how to make performance aids sensitive to human characteristics.

A.1.1. Overview of Approaches to Cognitive Modelling

11. There are a variety of approaches to modelling cognition that focus on different types of structure or process. For example, some emphasise the way in which objects and concepts are stored (how knowledge is represented), whereas others concentrate on the reasoning processes used (e.g., formal logic, everyday reasoning).

12. An early and still dominant viewpoint is to describe cognition in terms of an information processing framework consisting of a series of processing stages. The central assumption is that cognition is based primarily on the manipulation and transformation of symbols which ultimately relate to things in the external world. The mind is a limited-capacity processor having both structural and resource constraints. One of the major limitations of the information processing framework is its restriction to bottom-up or stimulus-driven processing, that is, processing directly affected by input from the environment. Cognitive activity also involves top-down, or conceptually-driven processing, where expectations are driven by the nature of the stimulus and the individual's past experience and expectations. The two kinds of processing work hand-in-hand in all cognitive activities.

13. An alternative to the symbolic information processing framework has emerged in the last few years. The connectionist approach rejects the stage-by-stage assumptions of information processing, and is based on parallel processing principles. Connectionist approaches seem to offer a better mechanism for learning, serendipitous recall, and creative thought.

14. Cognitive structures and processes have been modelled using three main approaches. The first (focusing mainly on the structure of concepts in memory) is based on the notion that an association can arise between two concepts, due to their similarity, their difference (contrast) or the fact that they occurred together in time. These associations can be modelled in a semantic network as concepts and links between concepts with activation strengths on the links. The triggering of concepts causes linked concepts to trigger as well, through spreading activation. Processes in the network serve to change the activation levels of links.

15. A second approach is based on production systems which model knowledge in the form of IF...THEN rules that are retrieved and brought into working memory (the part of the system holding information that is currently being processed). The system operates by matching the contents of working memory against the IF parts of the rule and executing the THEN parts. Conflict resolution may be required to choose between rules whose IF parts are simultaneously satisfied. Productions system are particularly useful for modelling procedures.

16. Finally, connectionist networks, neural networks, or parallel distributed processing models are a third approach. They have no explicit rules and symbols, but use distributed, layered representations of interconnected nodes in which concepts are characterised as patterns of activation in the network. These networks can learn to associate certain stimulus patterns with certain response patterns, even though no rule for the correspondence exists.

17. As can be seen from this brief description, there is quite a range of theoretical approaches to cognition, but there are similarities. The next section provides more detail on the common elements.

A.1.2. Fundamental Characteristics of Cognitive Entities

18. Several researchers have proposed basic principles of cognition. Newell et al. (1989) proposed that any cognitive system must have (at least) the following capabilities:

a) be able to behave flexibly as a function of the environment;
b) exhibit adaptive (rational, goal-oriented) behaviour;
c) operate in real time;
d) operate in a rich, complex, detailed environment, using vast amounts of knowledge and controlling a motor system of many degrees of freedom;
e) use symbols and abstractions;
f) use language, both natural and artificial;
g) learn from the environment and from experience;
h) acquire capabilities through development;

i) live autonomously in a social community;
j) exhibit self-awareness and sense of self.

19. Norman (1985) defined the elements that are common among cognitive systems:
a) a way of receiving information about the world (receptors);
b) a way of performing actions on the world (motor control);
c) a means of interpreting and identifying information received by the receptors;
d) a means of controlling the actions to be performed;
e) a means of guiding the allocation of cognitive resources when more needs to be done than can immediately be done;
f) a memory for the history of actions and experiences. (p 321)

20. Norman reasoned that

> "...because resources are finite, there will be times when more is being attempted than can be accomplished; some means of resource allocation (attention) will be required;
>
> because there will synchronisation problems with events in the environment and internal events, buffer (short-term) memories are required.
>
> There must be basic operations, an interpreter, and some feedback mechanisms that can observe the effect of operations upon the world and change accordingly.
>
> There must be some way to devise plans and then to monitor their operation; this requires levels of knowledge - meta-knowledge.
>
> For intelligent interaction, there must be a model of the environment, of one's self, and of others.
>
> There must be learning, changing one's behaviour and knowledge in fundamental ways (as opposed to simple adaptation), and this will probably require a system capable of inferring causality, inter-relations among concepts and events, and self-observation." (p 321)

21. Cohen (1989) points out that memory must also be selective and dynamic; it must link past, present, and future; it must be able to construct hypothetical representations; it must store both general and specific information; and it must store information implicitly.

22. Norman (1985) noted that "animate systems" have goals and desires and that these motivations impact the regulation of the amount of effort put forth for a given task. Humans' goals and values have been rather neglected concepts in theories of cognition. Goals can be directed to the immediate future or to the longer-term. The latter coincide with our beliefs and our world knowledge that undoubtedly have a large influence on our everyday reasoning.

23. Humans exhibit further characteristics and limitations that derive from their cognitive makeup. Some examples are the following:
a) Attentional seriality, requiring mechanisms for handling attentional focus. Especially important is the balancing of flexible adaptation to the environment and coherent attention to goals (Simon, & Kaplan, 1989, p 38).
b) Cognitive economy; for example, the partitioning of the objects and entities in the world into classes, thus decreasing the amount of information that has to be perceived, attended to, learned, remembered, communicated, and reasoned about (Rosch, 1978; Smith, 1988).
c) A propensity for structuring the world in terms of concepts which "... enable us to go beyond the information given" (Bruner, Goodnow, & Austin, 1956).
d) Concepts act as recognition devices; they serve as entry points into our knowledge stores and provide us with expectations that we can use to guide our actions.

e) The ability to combine concepts to form more complex concepts and thoughts (Smith, 1988, p 19-20).
f) The use of default reasoning, whereby properties of a class are taken to be true of its instances unless they are explicitly marked to the contrary (Quillian, 1966).
g) The use of approximate and inexact models o: the world, since for most purposes, it is not necessary to know precise numerical values. We need to be able to make rough estimates and fairly crude relational judgements. Approximate or relational knowledge is also more economical in storage. (Cohen, 1989, p 142)

24. Research on cognitive performance has identified its basic characteristics and has suggested why cognition is less than perfect, why it does not involve an exact representation of the world. The foregoing are general principles of cognitive behaviour that are common to all views of cognition. The principles of cognition are important concerns for COADE because they lead and shape the cognitive analysis.

A.2. ELEMENTS OF THE COGNITIVE SYSTEM

25. The models developed by the research community to explain and explore cognition have typically focused on different aspects of it. This section identifies and discusses the elements that are common across the various theoretical viewpoints. We use the following structure (derived from the above) as a basis for organising the discussion of the elements of cognitive entities:

a) Receptors - for receiving information about the world;
b) Motor control - for performing action upon the world;
c) Memory & Representation - for storing the history of actions and experiences; for buffering the reception of information ar he effecting of actions
d) Basic Processes - fo: identifying, interpretin d reasoning about information; for controll z actions; for modelling the env nment and self; for devising and monitoring ans;
e) Meta-cognitive Processes - for controlling the allocation of attention and cognitive resources;
f) Learning Processes - for updating memory.

26. The remainder of this section will address the core elements of the cognitive system, omitting receptors and motor control mechanisms. The following discussion draws heavily from the overview of cognitive psychology given by Eysenck and Keane (1990), from the extensive review of cognitive science in Posner (1989), and from the treatment of memory in Cohen (1989). These references are recommended as further reading.

27. We begin with a consideration of memory, and follow with a description of the representation of knowledge. The next part discusses the basic operations involved in cognition by reference to the primary theories in cognition, those of logical reasoning, decision making and problem solving. This is followed by a consideration of the special processes in meta-cognition and in learning. We then present a working model of cognition and problem solving that is based on the concept of the schema.

A.2.1. Memory

28. Human memory plays a central role in cognitive activity. Information from the external world is encoded and stored either temporarily or permanently in memory. Concepts and procedures are retrieved from or developed in memory, and memory is re-structured. Thus, both the structure of memory and the processes that act on it are important. Memory overlaps with the representation system, but can be considered to be a broader view of the collection, the encoding, and the retrieval of the representations. "The study of memory involves the demonstration that behaviour has been altered as a consequence of the previous storage of information.... There is an intimate relationship between memory and learning." (Eysenck, 1990, p 217)

29. A well-established theory of memory holds that the architecture is organised in terms of

modality-specific iconic memories, a short-term memory (STM) of limited capacity and a long-term memory (LTM) of unlimited capacity and duration. Information from the environment is received first in the sensory store, where it is held very briefly. If it is deliberately attended to, it is transferred to STM, which has a very limited capacity (about 8 digits). Information in STM is easily displaced by interference and diversion of attention. Long-term memory holds information from the "psychological past" that is not currently being processed. Transfer of information to LTM can be accomplished by rote rehearsal, with a direct relationship between the amount of rehearsal and the stored memory trace.

30. The ease with which information is retrieved from LTM appears to be a function of the degree and manner in which it has been processed or encoded at the time of learning. Deeper levels of analysis (in terms of the "meaningfulness", or semantic content extracted at the time of learning) produce longer-lasting and stronger memory traces. The degree and nature of elaborations on the information (made at the time of encoding) as well as the distinctiveness of processing (even at the non-semantic level) affect retention. The nature of the retrieval cues is a factor: Cohen (1989, p 122) points out that "the order of efficacy of the retrieval cues when presented singly was what, where, who, when. What was by far the most powerful cue and when was almost useless."

31. Humans have a better ability to recognise something in the external environment that they have seen before than to recall the same thing from memory. However, the success of recall is also a function of the similarity between the encoding at the time of storage and at the time of retrieval, a relationship known as encoding specificity (Tulving & Thomson, 1973).

32. Recently, the concept of working memory (WM) has received support. The concept of WM has been typically used to refer to the part of memory that is has been brought temporarily into focus for a task (in the sense of a working buffer). However, a slightly different view of working memory (Baddeley & Hitch, 1974) regards it as an extended replacement for STM, involved with both transient storage and active processing of information. In this view, WM consists of an loop, holding information in a speech-based form; a visuo-spatial scratch pad, which is specialised for spatial coding; and a central executive resembling attention. Eysenck and Keane (1990) argue that some kind of working memory is required for complex cognitive tasks (e.g., problem solving, text comprehension) that involve a number of different processing stages, since it allows the "state of play" of the task to be stored and readily available for further processing.

33. The connectionist view does not differentiate between types of memory, and indeed, regards processing and knowledge structures as tightly coupled (Rumelhart, 1989). In the connectionist model there is a set of linked processing units (perhaps representing conceptual objects like words, features, or larger concepts). The strength of the linkage is called the weight. Units accept input from and provide output to connected units according to activation rules that incorporate weights. This results in a state of activation for the units in the net. The pattern of connectivity constitutes the "memory" of the system. Changing the knowledge structure ("learning") involves modification of the patterns of connectivity, through development of new connections, loss of existing connections or modifications of the strength of connections that already exist.

34. It is generally accepted (Eysenck, 1990) that there are real distinctions between long term and working memory. The distinction between other types of memory can be useful, but there is still an open question whether there are different neural mechanisms for these other types of memory.

A.2.2. Representation of Knowledge

35. Representation of knowledge is necessary because as living beings we must interact with the world, with others, as well as monitoring ourselves. In order to survive and thrive we must have some means to perceive what we sense, to symbolise what we learn, to formulate what we intend to do, to monitor these actions, to make corrective actions, and to apply what we know to different situations. We must have a representation system so that we can think before we act, make plans, and test the plans without taking irreversible actions.

A.2.2.1. Types and Organisation of Knowledge

36. One useful distinction that can be made concerning types of knowledge is between declarative knowledge and procedural knowledge. Declarative knowledge is knowledge of facts (on concepts and relationships) — knowing "that" something is the case. Procedural knowledge is knowledge about "how" to do something (like ride a bicycle)[5].

37. Another heuristically useful distinction is based on a division between memory for semantic knowledge and memory for events or episodes (Tulving, 1972). Semantic memory refers to memory for facts about the entities and relationships between entities in the world (e.g., that birds have wings and that a canary is a bird). In contrast, episodic memory refers to events associated with personal experiences that happened at a particular time and place, and thus has an "autobiographical" flavour. Semantic knowledge is derived from episodic memories by processes of abstraction and generalisation. Episodic memories are interpreted and classified in terms of semantic knowledge in the form of schemata and scripts. (Cohen, 1989, p 114-115)

38. Current theories on the mental organisation of knowledge can be divided into three lines. The lines represent primarily a division of research perspectives, rather than a division in actual subject matter. The first category of theories deal with how different entities come to be grouped together under a common concept (e.g., all four-legged furry house pets that bark are categorised as 'dogs') and how these categories are related to one another hierarchically. Here the theories concern object concepts and (to a lesser degree) relational concepts.

39. The second category of theories deals with the structuring of knowledge about everyday events and the organisation of sequences of those events into plans.

40. The first two categories of theories assume that knowledge can best be represented in a language-like structure that is organised by rules (a propositional structure). A third category of theories suggests that some kinds of knowledge are represented in the form of mental analogues or images of the real world. Knowledge involving the spatial relationship between objects (e.g., geographical knowledge) is one example. Hybrid representations consisting of a mixture of analogical and propositional forms have also been proposed.

41. The following sections describe in more detail the concepts associated with representation of objects and events, and then concludes with a discussion of hybrid models and a broader kind of representation involved in problem solving called mental models.

A.2.2.2. Knowledge on Concepts and Relations

42. Holyoak and Nisbett (1986, p 66-67) propose that concepts can arise as the result of bottom-up or top-down processes. Bottom-up category induction involves the detection of multiple correlated properties that cause the instances with those properties to stand out as a natural class, distinct from other categories. Top-down triggering of a category, by contrast, is directed by the goals of the learner.

43. There are two principle views on how concepts and objects are organised. One view postulates that a set of attributes (or properties or features) can be defined for a concept. Two kinds of attributes are proposed. Defining attributes constitute the core definition of a concept. They are those attributes that are shared by all members of the category. The set of defining attributes is necessary and sufficient for determining whether a certain thing is a member of the concept category. (The intension of a concept consists of this set of attributes. The extension of the concept consists of all instances (or members) of the concept.)

44. Characteristic attributes, on the other hand, are those attributes that determine how typical or representative a member of the category is likely to be judged. According to this theory, the fundamental process of determining whether a entity is a member of a category (called feature comparison) is done in two stages: first, all the attributes of category and possible member are compared for overlap; if this fails to determine membership, then only the defining attributes are

5 This distinction has been useful in explaining skill acquisition in humans, where it is postulated that initially, what is learned is encoded declaratively, but with practice, it becomes compiled into a procedural form of knowledge. This process will be discussed in a subsequent section.

compared. Concepts can be organised into hierarchies where the defining attributes of a more specific category include all the attributes of its parent (through inheritance). These theories predict that category membership is quite clear-cut (as specified by the defining attributes); however, there is evidence that people consider the boundaries between categories to be fuzzy.

45. A more recent view on concepts and objects holds that they are organised around central prototypes which are represented by characteristic attributes. There are no defining attributes, and category boundaries are unclear. An object is considered to be an instance of a concept if its attributes match those of the prototype's attributes above some threshold. Thus instances of a category can be ranged in terms of typicality — some members are "more typical" than others. An alternative formulation of the prototype view represents the prototype in terms of the best member (or small set of best members) of the concept. Either way, the important assumption in this view is the idea that individual entities, not abstractions, are the central component of concepts. There is some question about the generality of the prototype approach, as it has been shown that some abstract concepts do not exhibit prototype structure.

46. Some work has been done on determining the nature of relational concepts (in contrast to concepts about objects). Many theories propose that relations can be characterised as set of relational primitives (e.g., primitive actions like transfer, propel, move) which take various objects as their arguments (e.g., the actor, the object acted on, the direction) in a case-grammar format (e.g., Schank, 1985b). Rather than this defining-attribute view of relations, though, recent research has favoured a prototype view, in which typicality of the action plays a large role in determining whether a particular relation is a member of a category.

A.2.2.3. Knowledge of Events and Situations

47. Concepts are related together in ways that reflect the temporal and causal structure of the world. Another line of research, concerned with the comprehension of complex events, has examined this form of organisation. The most commonly-used construct to account for complex knowledge organisation is the schema.

48. A schema is a cluster of information describing the characteristics of a situation, together with procedural knowledge about the typical procedure, sequence of events, actions or solutions associated with the situation. Thus the schema provides a structure for linking declarative knowledge based on experience (possibly in the form of a concept network) with a procedural network that gives the actions and sub-actions necessary to achieve the goal associated with the situation. So, one part represents information about stereotypical situations (and thus assists in recognising or describing problems and situations). The other part gives the typical sequences of events, actions or solutions associated with the situation. Theoretically, the concept of schema is a loose one in many respects, but it is widely accepted and has appeared in several forms, including story grammars, scripts, and frames.

49. Functionally, schemata consist of various relations, variables (or slots), and values. The relations can take a variety of forms, e.g., simple membership; actions like hit, or kick; or more complex causal relations like "prevent". Variables contain concepts or reference other schemata. Values refer to the various specific concepts that fill or instantiate slots. However, slots can be left open or filled with default concepts that are assumed if the slot is unfilled.

50. One well-developed variant of schema theory postulates the existence of scripts (Schank, 1985a). Scripts are sequences of actions that are temporally and causally ordered and that are goal-directed. Scripts represent knowledge about generalised, frequently-executed events (for example an individual may have scripts for eating, firing a weapon, etc.) Scripts provide an organising framework to allow us to infer facts about a situation that were not explicitly stated and to guide the comprehension and remembering of events (Cohen, 1989).

51. Later extensions of script theory addressed the inflexibility (and lack of economy in storage) of pre-determined scripts by proposing that scripts are created dynamically, as needed, from Memory Organisation Packets. MOPs are a kind of generalised high level script that represent the event sequences common to different situations (e.g., enter building, take seat, wait ... is a sequence common to cinemas, restaurants, doctor's visits, etc.). Within this memory system a particular episode, like

going to a party, can therefore be stored at several different levels of generality, such as: Going to David's party last Saturday evening; or Going to parties; or Social interactions (Cohen, 1989, p. 115-116). The MOPs are selected and combined as required under the control of a hierarchical goal structure called a plan that is tailored to the needs of the situation.

52. Plans stored in memory can be considered to be part of prospective memory, which is memory that pertains to events in the future. The term contrasts with retrospective memory, memory for events experienced in the past. In addition to knowledge about what to do (i.e., the contents of the plan, which may be stored as a hierarchy of actions), prospective memory holds information about the priority and timing of plan implementation. The encoding of information for prospective memory differs from that of retrospective memory, since prospective plans are usually self-generated and do not involve initial learning. With retrospective memory the test of success is the accuracy and completeness of recall; with prospective memory the test of success is the execution of a plan that is appropriate and timely (Cohen, 1989).

53. Non-standard aspects of a particular occasion are stored as specific pointers, tags or indices, which serve to retrieve the memory of that event. Schank (1985b) postulates that these specific and unique pointers provide the mechanisms for 'reminding'.

54. There is considerable evidence from psychological studies in several areas for the operation of schema-like knowledge structures. Schemata have been used to account for human ability to make inferences in complex situations, to make default assumptions about unmentioned aspects of situations, and to generate predictions about what is likely to occur in the future.

A.2.2.4. Hybrid Representations and Mental Models

55. Research on representation of knowledge suggests that some kinds of information may be represented analogically, that is, in a form similar to an image or model of the object in the real world. Analogue representations are modality-specific (i.e., visual, auditory, tactile, olfactory, kinetic) images. In contrast to propositional representations, which can be characterised as being abstract, discrete in nature, and organised by strict rules, analogical representations are concrete, nondiscrete, and organised by loose rules of combination. They are dynamic, in the sense that they can be rotated and transformed, thus making them especially suitable for tasks using spatial information.

56. There has been considerable controversy over the role and importance of analogical representation, and its relationship to propositional forms of knowledge representation. Hybrid models using both types of representation have been proposed by Kosslyn (1980) and others. According to Kosslyn, analogue spatial images of scenes or objects are constructed as temporary displays from deeper propositional knowledge in LTM.

57. A broader and more comprehensive knowledge representation mechanism is the mental model. According to Johnson-Laird (1989), mental models are working models of real world situations, derived from perceptual and verbal information, and constructed when required. They can represent both physical relations, space, movement, etc.; and conceptual features such as negation. Even if they are incomplete and unstable (and sometimes inaccurate), they are used as a basis for simulating hypothetical actions and events, and for generating predictions and explanations. This makes them especially well-suited for planning, and for problem solving in novel situations, especially where the problem is ill-defined.

58. The exact difference between mental models and schemata has not been theoretically clarified. A potentially useful distinction might be that schemata (knowledge structures about a specific domain) can be either abstract or instantiated with knowledge based directly on episodic experience. In cases where no instantiated schemata applicable to the domain of the problem exists, humans can still use schemata to imagine general causal connections that could exist. The structure that results from the modelling of the connections is the mental model.

59. Finally, there is an even broader representation concept, that of problem space. The problem space is the set of information pertinent to the problem at hand, including the representation of the objects in the problem situation, the goal of the problem, the actions (operators) that can be performed to transform the representation. It includes, as well, strategies that can be used in working on the

problem and a knowledge of the constraints in the problem situation. (Greeno & Simon, 1986, p 591)

A.2.2.5. Summary

60. Some of the terms used for describing representation address a specific, single idea (e.g., propositions), while other terms relate to more complex structures of knowledge (e.g., concepts, mental models, and schemata). Representations are related to varying degrees to one another and can be associated or differentiated on the basis of other aspects, commonly referred to as attributes, features, properties, and characteristics. Representations stand for objects, events, locations, beliefs, notions, memories, problem spaces, and anything else that is thought about.

61. Our concern about concepts for structuring knowledge must be tempered by a realisation that the knowledge held by a human is a unique version of what exists. It may be partial, different, misleading or incorrect.

62. In spite of the variety of definitions, the idea that knowledge exists as a mental representation is important to the goals of COADE, because it relates directly to how knowledge is elicited from people about a problem domain and how system support can be designed to aid performance.

A.2.3. Basic Processes and Operations in Cognition

63. Operations are the cognitive actions used to encode, retrieve, manipulate, and generate representations. This section will review some of the theories of cognition to indicate the nature of the basic processes or operations.

64. The situations in which cognitive or "thinking" activities are carried out vary enormously: both in the amount of knowledge used, and in the extent to which thinking is actually consciously and deliberately directed to a specific goal, or is more automatic. Rasmussen (1986) proposed a three level model of processing. The most automatic level of performance was referred to as skill-based. The next level is rule-based whereby rules are instantiated for a routine situation and the procedures carried out. In novel situations, reasoning is knowledge-based. Here the problem solver operates only with the declarative knowledge and must use general inference procedures. These levels of processing also represent continua of automaticity, required attention, and deliberate analytical procedures. They have been used to help distinguish among types of error (Reason, 1990) and are discussed in more detail in the sections on errors and limitations (Sections 5.1.2.3., 5.1.4.1., and 6.4.3.).

65. There are two main views on how humans think. One is based on language and logic and regards thinking as a process of inference or reasoning, usually using a language-like representation (Rips, 1986). The other is based on the notion of heuristic search for problem solutions generally using representations that model the problem situation (Simon & Kaplan, 1989). There is evidence that these two perspectives are merging.

66. A distinction is often made between deductive reasoning and inductive reasoning. When people carry out deductive reasoning, they determine what conclusion can be drawn when certain statements (propositions) are assumed to be true. Reasoning proceeds from the general to the specific. In these circumstances, no increase in semantic knowledge is achieved. In inductive reasoning, people make a generalised conclusion from premises that describe certain instances. Thus, inductive reasoning involves inferential processes that expand one's knowledge (through the generation of hypotheses to be tested) in the face of uncertainty.

67. Logic, and in particular, propositional calculus, has been used to characterise the abstract structure of many deductive reasoning problems and to set the norms for performance. These studies involve the use of propositions related together by the conditional if ... then (e.g., if P then Q, where P and Q are propositions) from which inferences can be made. Several theories have been proposed to account for people's actual performance on these deductive reasoning tasks, a performance which can be regarded as rational but flawed (Eysenck & Keane, 1990). An important factor seems to be the way

in which the pragmatics of the situation enter into modelling of the problem premises from which the conclusion is drawn.

68. Theories on decision making have emerged principally from empirical studies on how people make statistical judgements, rather than everyday decision making tasks. In these studies, human performance is compared to norms generated from the statistically-correct (and efficient) ways of making a judgement (usually involving a probabilistic choice between the utility of several outcomes). Human behaviour has been shown to deviate from the optimal (or normative) decision approach, and these deviations have been classified as biases in decision making. Two of these biases are the "availability heuristic" and the "representativeness heuristic". The availability heuristic refers to the tendency to use the availability of instances or scenarios in assessments of the frequency of a class or plausibility of a happening. The representativeness heuristic makes use of the similarity of a target item to a representative instance of a class in making a judgement about the probability that an object or event belongs to another class or process. (See Section 6.4.2.3 for additional discussion of biases.)

69. Early theories on problem solving were rooted in the Gestalt approach and considered problem solving to be both reproductive and productive. Reproductive problem solving involves the re-use of previously-learned techniques acquired by experience; productive problem solving is characterised by insight (which can come suddenly) into the structure of the problem and by restructuring of it.

70. One widely-held view on problem solving is that based on the notion of a problem space. The fundamental idea is that the objective structure of a problem can be characterised as a set of states, beginning with an initial state and ending with the goal state (i.e., the problem is solved). In between, there is a whole set of possible states and paths connecting states, some of which lead to the goal and some of which do not. The key thesis in the problem-space theory is that people solve problems by applying mental operators to move between states, searching for the pathway to the goal state. Since a given problem may have a large number of alternative paths, people use strategies (or heuristic methods) in order to guide the search process and to move efficiently from the initial state to the goal state. The various intermediate states are held in working memory while long-term memory holds the set of productions and operators that modify the states, and so problem solving behaviour is determined partly by the limitations of the cognitive architecture (e.g., working memory capacity).

71. Heuristic methods are distinct from algorithms. Use of the latter will ensure the generation of a solution to the problem. Heuristic methods are more like "rules-of-thumb" that do not guarantee a solution, but are likely to, and can save time and effort. One of the most important heuristic methods for reducing the number of alternative states that must be considered in problem solving is means-end analysis. It consists of noting the difference between the current state and the goal state, the creation of subgoal to reduce the difference, and the application of an operator to solve the subgoal. This suggests that the ability to structure a problem is an important factor in solving it efficiently, and indeed, humans improve in their ability to do this with increasing experience.

72. VanLehn (1989) has proposed a more detailed schema-based[6] theory of problem solving, in which two basic processes work in a complementary, and possibly interleaved manner. The understanding process generates an internal representation of the problem (represented by a set of assertions) and the search process operates on this representation. Understanding involves the subprocesses of assimilating the problem, and converting the premises into the internal representation needed by search. In well-defined, knowledge-lean tasks understanding is a rather direct translation process, governed mainly by the type of stimulus and the need for an internally-consistent problem space. For knowledge-rich problems, the understanding process is more complicated, possibly involving a subprocess of elaboration, where new assertions are added to the start set, without removing any of the old assertions or decreasing their potential relevance. Search is the process responsible for finding or calculating the solution to the problem. This process involves small, incremental changes to the problem state, through the application of operators. Search is guided by a proceed strategy (that determines which operator will be chosen for application to the problem) and by a backup strategy (to allow for backtracking to a previous state when the application of an operator is unsuccessful).

---

6 A fuller account of schema-based problem solving will be given in a subsequent section.

73. The problem space theory has been quite successful in predicting the behaviour of humans in solving well-defined problems (like puzzles), where the operators, the initial and goal states are well-specified and little domain-specific knowledge is involved. In these cases, domain-independent (or "weak") heuristic methods can be made to work (although not always efficiently). In contrast, ill-defined problems which are underspecified (in terms of initial and goal states) and require the use of substantial amounts of domain-specific knowledge are not addressed by this theory, although it may provide a good basis for predicting performance on these types of problems too.

74. Some theorists propose that prior experience in other domains may assist problem solving in a different kind of way by providing analogies that can help in a problem-solving situation. This analogical thinking has been characterised as being the result of processes that transfer or map the conceptual structure of one set of ideas (called the base domain) to the current (target) domain. Both concepts and relations can be mapped. A key aspect of analogical thinking is that the transfer of knowledge typically involves coherent, integrated pieces of knowledge, rather than fragmentary bits. Once a good analogy is established, the base solution may transfer directly, and holistically, to the target. The degree of similarity between the target and base domain is one factor determining the rate of spontaneous analogical transfer (Holyoak & Koh, 1987). Although surface similarity is the best predictor of memory access (in forming the analogy), similarity in relational structure is the best predictor of ratings of inferential soundness (Gentner & Landers, 1985). Once an analogy has been developed, general rules are formed from it, so that later cases of the problem are classified directly by those rules (Holyoak & Nisbett, 1986).

75. These operations are examples of transformations based on comparisons, combinations, and new constructions. Cognitive operations are important in COADE because they lead to interpretations, conclusions, problem solutions and decisions which are the focus for computer-based aiding.

A.2.4. Meta-cognition

76. Meta-cognition can be considered as "knowledge or beliefs about one's own cognitive processes" (Eysenck, 1990, p 225), that is, the awareness that one's own cognition can be used to manage one's thinking. Meta-cognition may play a role in setting goals, selecting strategies, organising thoughts, controlling, search, allocating attention, self-reflection (for assessment of performance), assessing likelihood of knowing, directing search, etc. Greeno and Simon refer to a related concept, strategic knowledge, as the process "for setting goals and adopting general plans or methods in working on a problem" (1986, p 590).

77. Cavanaugh (1988) describes three kinds of memory awareness. The first is systemic awareness, which is knowing how memory works, what kinds of things are easy or difficult to remember, or what kinds of encoding and retrieval strategies produce the best results. Epistemic awareness is knowing what we know, knowing what knowledge is in store, and being able to make judgements about its accuracy. (It is also called meta-knowledge). The third kind is on-line awareness, that is, knowing about ongoing memory processes and being able to monitor the current functioning of memory, as in prospective memory tasks. Cases of absent-mindedness occur as result of failures of on-line awareness (Cohen, 1989, p 142-143).

78. One meta-cognitive process is associated with the ability that people have to selectively attend to chosen stimuli and ignore other stimuli that are present. This is called selective attention The stimulus channels must be different in some physical way (loudness, location, etc.). The stimuli being ignored may be distracting when certain signals are included that attract attention.

79. There is also evidence that awareness of appropriate strategies for remembering (e.g., phone numbers or dates of events) increases with age. This phenomenon has been termed meta-memory. The pattern of age effects in old and young suggests that meta-memory efficiency improves with experience but is unaffected by the size of the knowledge base (p. 149, Cohen, 1989).

80. Related meta-cognitive issues have been examined to provide insight into cognition and memory. One is the feeling of knowing (and its converse, feeling of not knowing). Studies by Read

and Bruce (1982); Gruenberg and Skyes (1978); Lachman, Lachman, and Thornsberry (1981) have found that although people cannot always judge what they will be able to retrieve from memory, there is a gradient of knowing which can be measured subjectively by feeling of knowing and confidence ratings, and objectively in the speed and accuracy with which a target piece of information can be retrieved (Cohen, 1989, p 146). One's sense of the likelihood of retrieving information is an important factor in allocating attention and selecting particular strategies (e.g., 'should I try to remember some conclusion or retrace my thoughts of deriving a conclusion or go look up what I wrote about it earlier?').

81. The use of meta-cognition is probably quite variable across individuals. Eysenck and Keane (1990) contend that, in fact, almost all of our cognitive processes take place without us being aware of what they involve or how they work. Knowing about meta-cognition can allow the COADE analyst to appreciate the potential shortcomings of what people report about their knowledge and thought processes. Also meta-cognition is a valuable construct for the decision aid developer. When decision makers are aware of specific weaknesses or preferred styles of remembering or strategies, they should be more likely to accept and use memory aids that are provided to them.

A.2.5. Learning

82. One of the essential characteristics of humans is that they are able to adapt through learning. Thus learning has a special role in cognitive science and is intimately connected with performance on tasks.

83. Simon and Kaplan (1989) suggest that there are three time frames over which humans adapt and learning occurs. In the short time scale, an intelligent cognitive entity solves each problem it encounters using different behaviour. In the longer term, the entity makes adaptations that are preserved and remain available for subsequent new situations. Information is accumulated in memories and access routes for retrieving it are also acquired. In the longest time scale, systems evolve both biologically and socially (p 38-39).

84. There are several theories that address the changes that occur within the second time frame. Rasmussen (1986) provides one theory about behaviour that is either formal knowledge-based, rule-based, or skill-based. Another similar theory was proposed by Fitts and Posner and then adapted by Anderson (1983) in his theoretical work on adaptive control of thought (ACT*). The model involves the declarative-procedural distinction and three stages. In the first stage, called the cognitive stage, problem solving is slow and tedious and prone to error; learning involves primarily the accumulation of declarative knowledge from various sources. It may also involve rule induction, where a particular sequence of moves is recognised as a pattern and a rule is induced to describe it.

85. As competency increases, the second stage of learning, the associative or procedural stage, begins. In this stage the repeated use of declarative knowledge in given situations results in domain-specific procedures, that is, direct associations between specific domain conditions and the resultant action. These direct linkages permit the decision maker to bypass the longer and more tedious process of retrieving declarative knowledge and applying general productions to it (Charness & Campbell, 1988). This stage and the next involve instantiation, the filling-in of a general production with declarative knowledge. This process of proceduralisation results in a gradual build-up of task-specific productions from general-purpose ones.

86. In the third (autonomous) stage, the procedures become increasingly specialised and automatised, through the strengthening of their associations to particular types of situations. Simple productions become composed into, or replaced by, more complex, inclusive productions in a process called compounding. Learning also occurs through a process of tuning, the modification of the operator selection heuristic, based on new experience; from a production standpoint, the condition associated with choice of the heuristic needs to be altered, either by making it more general (generalisation) or more specific (specialisation)."When performance of a task has become completely automatised, processing requires virtually no cognitive resources, is autonomous, and is unavailable to conscious awareness." (Gordon, 1992, p 100-101)

87. The description of schema-based thinking in the next section and the discussion of the

elements of cognitive models above have implications for how adaptation occurs during the shorter time frame. Learning typically occurs during an "impasse" in processing. The next section describes how schemata are "repaired" when an impasse is encountered.

88. It is important for the COADE analyst to recognise that an essential characteristic of humans is their capacity to adapt and learn. Learning brings about change in performance over time. This tendency for continual change must be kept in mind during the cognitive task analysis, design, evaluation, and planning for the implementation of decision aiding. People may change in subtle ways and surpass the capabilities of the aid that is delivered. People may also develop new or more sophisticated strategies based on self-discovery during knowledge elicitation.

A.3. SCHEMA-BASED PROBLEM SOLVING

89. Schema-based problem solving was chosen as the modelling concept to discuss in more detail for several reasons. Schema models easily relate to thinking about everyday problem solving. There is a fair amount of research that has explored and defined implications from schema-based models. Schema-based problem solving seems to characterise experts who are solving problems in knowledge-rich domains (VanLehn, 1989). Schema-based models are suitable for incorporating other cognitive modelling concepts. Furthermore, the solution procedures of schemata are likely to correspond to the products of learning mechanisms (VanLehn, 1989), thus making a necessary link between problem solving and learning.

90. Schema theory proposes that new experiences are not just passively 'copied' into memory. Instead, a memory representation is actively constructed by processes that are strongly influenced by schemata (Alba & Hasher 1983) as described by Cohen (1989, p 72):

a) Selection: The schema guides the selection of what is encoded and stored in memory. Information that is relevant to whichever schema is currently activated is more likely to be remembered than information that is irrelevant.
b) Storage: A schema provides a framework within which current information relevant to that schema can be stored.
c) Abstraction: Information may undergo transformation from the specific form in which it was perceived to a more general form. Specific details of a particular experience tend to drop out, whereas those aspects that are common to other similar experiences are incorporated into a general schema and retained.
d) Normalisation: Memories of events also tend to be distorted so as to fit in with prior expectations and to be consistent with the schema. They are sometimes transformed toward the most probable or most typical event of that kind. People may remember what they expected to see rather than what they actually saw.
e) Integration: According to schema theory, an integrated memory representation is formed which includes information derived from the current experience, prior knowledge relating to it, and default values supplied by the schema.
f) Retrieval: Schemas may also aid retrieval. People may search through the schema in order to retrieve a particular memory. When the information that is sought is not represented directly, it can be retrieved by schema-based inferences. (If you know that John has measles, you can infer, from your measles schema, that he won't come to the party.)"

91. VanLehn suggests that the solving of routine problems is a fairly direct process that makes extensive use of schemata (thus permitting the explicit search stage to be bypassed). He proposes that there are three main processes involved. The first is schema selection, where a potentially-usable schema is triggered, often early in the processing of the problem stimuli. The triggering process is not very well understood. The schema guides interpretation of the rest of the problem by setting up expectations that direct the search for more specific schemata.

92. Selection of a schema goes hand in hand with adapting it to the given problem, that is,

filling in the slots that make the schema match the parameters of the situation. This process is called instantiation. Often occasions of slot filling are mingled with occasions of specialisation, where a schema is rejected in favour of subordinate schema.

93. The third and final stage in schema-based problem solving, once a schema has been selected and instantiated, is to follow the solution procedure that is suggested by the schema itself.

94. There are cases in which schema selection and instantiation are not routine and straightforward. One case is when several schemata are triggered by a situation. In this case, the decision maker may tentatively choose one, but must be prepared to back up and re-select in the case that it is not usable (VanLehn, 1989) In the solving of non-routine problems no individual schema completely addresses the entire problem, yet several schemata may address specific aspects. The decision maker must join and integrate the relevant schemata so that they cover the whole problem (a process called schema compounding). In other nonroutine cases, an "impasse" can be encountered. This happens when the schema describes a condition which does not match with the current state or when the action called by the schema is infeasible for the situation. The ensuing process is called "repairing," which may involve fixing the problem or, in many cases, rejecting the schema outright and selecting another.

95. Research on schemata has elucidated some of the everyday failures that people experience with memory. For example, people are sometimes unable to distinguish in memory between what they have actually perceived and what they have only heard about or imagined. This effect was explored in an experiment by Bre[illegible] and Treyens (1981), who concluded that schemata tend to normalise views that people have o[illegible]tuations, so that people remember the objects that they expect should be associated with the situation. Schemas also induce inaccuracies and distortions in memory for events. There is evidence that people confuse events from different scripts in memory. There seems to be memory structure (the MOP) that is common to multiple scripts.

96. Schema theory might lead one to believe that what is typical and consistent in existing knowledge will be remembered better than what is unexpected. But it seems that the unusual or unexpected is more salient, so that novel or atypical occasions seem to stick in memory. The schema-plus-tag model has been proposed to account for this. This theory suggests that the irrelevant, unexpected, or deviant aspects of an event are remembered easily and act as markers for the retrieval of the remaining aspects of the episode.

97. The recall of actions is influenced by the organisation of plan schemata. Brewer and Drupee (1983) showed that actions higher up in the hierarchy of plan schemata are better remembered than those lower down and that, over time, information about actions was lost first from the lower-most (detailed) part of the hierarchy and then gradually upward. However, the schema could be used to reconstruct the whole action sequence at the retrieval stage.

98. The means by which schemata are learned have not been much investigated. Three methods have been proposed: accretion, tuning and restructuring. In learning by accretion, one simply records a new instance of an existing schema and adds it to the repertoire. Tuning refers to the elaboration and refinement of concepts in the schema through experience (e.g., the discovery that a new type of object can fill a slot). Restructuring involves the creation of new schemata either by analogy or by induction.

99. These distinguishing characteristics of schema-based thinking and extensions of theories provide insight into potential strengths and weaknesses of thinking and memory. As the theories are further refined there will undoubtedly be more explanations for the typical characteristics of remembering and not.

100. Schema theory is important to COADE because it addresses important aspects of cognition, including ways in which cognition operates and ways in which the natural efficiencies lead to limitations in some cases.

A.4. IMPLICATIONS FOR COADE

101. The intent of this discussion of cognition and cognitive concepts was to familiarise the user of COADE with issues of cognition. With a better appreciation of what cognition is, the user of COADE can search out information and cognitive scientists to contribute to the analysis, design, and evaluation of decision aiding. The specific findings addressed in this chapter about the characteristics of memory, learning, attention, etc. can be used during COADE application. The sources upon which this chapter is based are recommended for more complete information. More specifically, the relevance of cognitive models for COADE activities includes the following.

102. The cognitive concepts provide a startpoint for cognitive task analysis. The elements of cognitive models have direct relationships to knowledge elicitation and knowledge engineering. Gordon (1992) observes that expert systems are moving in the direction of multilevel knowledge structures and can be influenced by the levels of abstraction and hierarchical knowledge structures seen in several of the theories.

103. For assessing cognitive performance, the COADE analyst must have a thorough understanding of cognitive limitations, workload capacities, and errors. The elements of cognitive models can be used as a general framework for error recognition: is a decision maker likely to have errors in the encoded memory structures, operational processes, allocation of attention (meta-cognition), or in adapting and learning to uncertain and unstable problem situations.

104. Cognitive models also influence the application-specific characteristics determined from the analyses of task, decision makers, and organisations. In the design activity, general cognitive categories are used in the structure of the decision aiding strategies.

105. This review began with an implication by Lewis that all designers of human-computer interfaces have a model that they use. The issue is not whether they use a model, but the extent that it is explicit or covert. The same implication applies to decision aid designers and analysts of decision making behaviour. By developing a better understanding by learning from and applying cognitive models, analysts and designers will be in a better position to develop decision aiding that work.

ANNEX I TERMS OF REFERENCE RSG.19

RESEARCH STUDY GROUP ON DECISION AIDS FOR COMMAND AND CONTROL

PROPOSED TERMS OF REFERENCE

I Origin of the Proposal

A. Background

1. A Command and Control Information System (CCIS) exists to assist the commander during all five phases of the Command and Control (C2) Cycle, i.e., maintaining status, assessment, planning, decision and execution. NATO CCIS systems have the additional complication of multinational communications and coordination. The command and control of battlefield operations makes heavy demands on the information processing and decision making capacity of the commander and his staff. The speed and sophistication with which computers can process and distribute battlefield data have obvious potential for alleviating this load. However, the complexity of the cognitive aspects of human decision making in the command and control cycle has meant that the full potential of computer aiding has not yet been realised.

2. NATO AC/243 (Panel 8/RSG.12) has evaluated the state of knowledge concerning aspects of computer human interaction in the command and control cycle, and has identified several important areas which require research.
These include:

(a) the development of tools and software environments for defining and prototyping the human-computer interface;
(b) methods for more efficient human factors evaluation of the user interface;
(c) development and integration of computer-based decision aids into C2 systems.

B. Military Benefit

3. The latter issue, decision aids, is particularly urgent in view of the number of computer-based decision tools that are being developed and promoted by various military and civilian laboratories within the NATO community. It is not clear, however, which ones have the most potential benefit in command and control systems. This is partly because of practical human decision making in actual command and control environments and about how these aids should be exploited and integrated.

4. Both NATO and the individual member nations would benefit from collaborative research and analysis which:

(a) brings together existing knowledge on human decision making characteristics in C2 operations; and
(b) assess, from a human viewpoint, existing and proposed decision aids.

II. Main Objectives of the Work

5. The objective of the RSG is to review the state-of-the-art of computer-based decision aids and to assess the potential impact of such decision aids on NATO C2 operations.

6. The major product will be a framework for the human factors evaluation of decision aids for command and control. An additional product of direct benefit to CCIS developers will be a review and assessment of representative decision aids and their suitability for decision support at various points in the C2 cycle.

7. The work of the RSG will be conducted by correspondence as well as through meetings, both at NATO locations and in the member countries. Demonstration visits to NATO and related command and control systems should form a substantial component of the work.

8. The duration of the RSG shall not exceed three years from its acceptance by Panel 8 unless extended by the Panel into a second phase.

III. Resources

(a) members of the group should have considerable knowledge of cognitive psychology or possibly human factors, combined with substantial recent experience of command and control systems;
(b) support from NATO would be required to assist holding a Workshop or ASI, probably in the second year; participation by STC would facilitate the possibility to draw some of the software to be identified by the RSG together on machines at STC.

IV. Security Level

9. For the exchange of some information, a security level of NATO SECRET will be required, although most of the work of the RSG will be at a NATO UNCLASSIFIED level.

V. Liaison

10. Maximum cooperation should be sought with other multinational bodies concerned with related problems, such as the Information System Working Group (ISWG) which operates under the NATO Command and Control Data Processing Committee.

ANNEX II RSG MEMBERS AND PARTICIPANTS

Addresses of RSG.19 members July 19, 1994

NL Dr. Peter J.M.D. Essens (chair)
TNO Human Factors Research Institute
P.O. Box 23
3769 ZG Soesterberg
The Netherlands
tel: [31](3463)56319
fax: [31](3463)53977
email: petere@tm.tno.nl

CA Ms. Carol McCann
Defence and Civil Institute
of Environmental Medicine
P.O. Box 2000
North York, Ontario
Canada M3M 3B9
tel: [1](416)635-2190
fax: [1](416)635-2104
email: cal@dciem.dnd.ca

FR Med. Princ. René Amalberti
Centre d'Etudes et de Rech.
de Médecine Aerospatiale
Base d'essais en vol
Bretigny sur Orge 91228 Cedex
France
tel: [33](1)6988-8781 ext. 2921
fax: [33](1)6084-0448

FR ISC. Jacques Maigre
DCAN/ INGENIERIE SUD
BP 30
03800 Toulon Naval
France
tel: [33]9416-2431/5209/4149
fax: [33]9416-2110

GER Mr. Gert Dörfel
FGAN/Forschungsinstitut
für Anthropotechnik
Neuenahrer Strasse 20
D-53343 Wachtberg-Werthhoven
Germany
tel: [49](228)852-474
fax: [49](228)[illegible]-953
email: doerfe[illegible]atvax.fat.fgan.de

USA Dr. Jon J. Fallesen (secretary)
ARI Field Unit Leavenworth
Bldg 90, McClellan Avenue
Ft. Leavenworth, KS 66027-0347
U.S.A.
tel: [1](913)684-4933
fax: [1](913)684-3290
email: fallesen@alexandria-emh2.army.mil

USA Dr. Gerald S. Malecki
Office of Naval Research
Cognitive & Neural Sciences Division
Code 1142
800 North Quincy Street, BCT #1
Arlington, VA 222175000
U.S.A.
tel: [1](703)696-4044
fax: [1](703)696-6887

STC SHAPE Technical Centre
Attn: Dr. Klaus Müller
P.O. Box 174
2501 CD The Hague
The Netherlands
tel: [31](70)314-2173
fax: [31](70)314-2111

IMS LTC H.G. Ippig
International Military
Staff - CIS Div.
NATO Headquarters
1110 Brussels
Belgium
tel: [32](2)728-5309
fax: [32](2)728-4117

USA Dr. Jan Cannon-Bowers
Naval Training Systems Center
Code 262
x12350 Research Parkway
Orlando, FL 328263224
U.S.A.
tel: [1](407)380-4830
fax: [1](407)380-4412

Other addresses:

NATO DRG Secretary
Dr. J. Vermorel
DRS-Defence Support Division
NATO Headquarters
1110 Brussels
Belgium
tel: [32](2)728-4759
fax: [32](2)728-4103

Invited participant
Major (USAF) T. Triscari, Jr
321 Ktahdin Dr.
Lexington, MA 02173
U.S.A.

Previous members:

Mr. Robert J. Kruchten
525 Brooks Road
Rome Laboratory/C23AA
Griffiss AFB, NY 13441-4505
U.S.A.

Mr. Hank C. Okraski
Naval Training Systems Center
12350 Research Parkway
Orlando, Florida 32826-3224
U.S.A.

Mr. Theodor W. Schlickmann
SHAPE Technical Centre
Postbus 174
2501 CD DEN HAAG

ANNEX IV TOR AD HOC GROUP

Date: 20 April 1994

AD HOC GROUP ON
THE APPLICATION AND EVALUATION OF COADE
(Framework for Cognitive Analysis, Design, and Evaluation)

PROPOSED TERMS OF REFERENCE

I. ORIGIN

A. Background

1. RSG.19 produced a framework for cognitive analysis, design, and evaluation (COADE). An initial version of COADE has been reviewed by a group of experts in a workshop (Workshop on Decision Aiding in Command and Control, June , 1992). The value of a framework such as COADE was supported by the experts. They recommended to treat the development of COADE as an evolutionary development. Experience from the application of COADE should be used to augment the framework.

2. In the development of Command and Control Information Systems there is in general little attention for cognitive aspects of the planning and decision making tasks. In a survey performed by RSG.19 most of the interviewed developers indicated that they would use a framework that addresses cognitively-centred system development. COADE provides a framework for addressing complex cognitive tasks.

II. OBJECTIVES

1. To gather feedback from experts on the new version of COADE.
After the workshop, where an initial version was reviewed, new material was added which makes it opportune to evaluate COADE again with experts.

2. To evaluate the applicability of COADE in actual applications.
COADE, as described in the final report of RSG.19, should be seen as a prototype that requires a 'field trial' to establish its usability, and effectivity. A first trial should be done by the COADE developers themselves. Eventually, a field trial has to involve a developer not involved in the development of the framework (Independent Validation and Verification). It is proposed that for the Ad Hoc Group the COADE developers themselves apply COADE in current projects.

3. The products of the work are:
   i. reviews by experts
   ii. a report on the application of COADE
   iii. an augmented COADE if required

4. The duration of the Ad Hoc Group will not exceed two years. Upon acceptance of the TOR by the Panel, the individual members will start analysing selected projects for application of COADE. Startpoint of the duration of the group will be the first formal meeting of the group.

III. RESOURCES

1. Membership The same members as RSG.19 are invited to participate in the Ad Hoc Group. The benefit of having the same members is that everybody is current with the concepts. A small group is preferred. However, the Ad Hoc Group is open for who is interested and committed to study COADE. In particular, people with experience in decision aid development are invited to participate.

IV. Security level: NATO SECRET.

V. Liaison. Contacts will established with the potential RSG's on Human Error and Cognitive Task Analysis.

ANNEX III WORKSHOP PARTICIPANTS

Dr. Leonard Adelman
Department of Systems Engineering
and C3I Center of Excellence
School of Information Technology and
Engineering
George Mason University
Fairfax, VA 22030
U.S.A.

Dr. Marvin S. Cohen
Cognitive Technologies, Inc.
4200 Lorcom Lane
Arlington, VA 22207
U.S.A.

Dr. Gary A. Klein
Klein Associates, Inc.
582 E. Dayton-Yellow Springs Road
Fairborn, Ohio 45324-3987
U.S.A.

Dr. William B. Rouse
Search Technology, Inc.
4725 Peachtree Corners Circle, #200
Norcross, Georgia 30092
U.S.A.

Dr. Andrew P. Sage
School of Information Technology
and Engineering
George Mason University
Room 100, 4400 University Drive
Fairfax, VA 22030-4444
U.S.A.

Dr. Martin A. Tolcott
3001 Veazey Terrace, N.W.
Apartment 1617
Washington, DC 2008
U.S.A.

Dr. Wayne Zachary
CHI Systems
Gwynedd Plaza III/Bethlehem Pike & Sheble
Lane
Spring House, Pannsylvania 19422
U.S.A.

Invited participant:

Mr. David Smith
US Naval Command, Control and Ocean
Surveillance Center
Decision Support and Artificial
Intelligence Branch
Code 444
San Diego, CA 92152-5000
U.S.A.